U.S. Government Web Server Protection Profile for Basic Robustness Environments

Version 0.41

Information Assurance Directorate

1 August 2003

Protection Profile Title:

Web Server Protection Profile (PP) for Basic Robustness Environments

Criteria Version:

This Protection Profile (PP) was developed using Version 2.1 of the Common Criteria (CC) [1] and applying the NIAP interpretations that have been approved by TTAP/CCEVS Management as of July 10, 2002.

Constraints:

Targets of Evaluation (TOEs) developed to satisfy this Protection Profile shall conform to CC Part 2 and CC Part 3 and applicable NIAP approved interpretations.

Table of Contents

List of Figures

List of Tables

1.0. INTRODUCTION

This Web Server Protection Profile (PP) for Basic Robustness Environments was sponsored by the National Security Agency (NSA). This Protection Profile is intended to be used as follows:

- For product vendors and security product evaluators, this PP defines the requirements that must be addressed by specific products as documented in vendor Security Targets (STs).

- For system integrators, this PP is useful in identifying areas that need to be addressed to provide secure system solutions. By matching the PP with available STs, security gaps may be identified and products or procedures may be configured to bridge these gaps.

1.1. Protection Profile Identification

Title: Web Server Protection Profile for Basic Robustness Environments Version 0.41

Authors: U.S. Government and industry

Vetting Status:

CC Version: 2.1

Evaluation Assurance Level (EAL): 2 Augmented

General Status:

Registration:

Keywords: Web, Server, HTTP, and HTTPS.

1.2. Protection Profile Overview

This profile specifies the minimum security requirements for a web server (hereafter referred to as the Target of Evaluation (TOE)) used by the United States Government in Basic Robustness Environments. The target robustness level of "basic" is further discussed in Section 3.0 of this PP.

The TOE is a software application that serves content via a specific set of Internet protocols in response to requests from a network. Some content is public and available to any requestor; other content has controlled access and must be protected from disclosure. Determination of whether content is public or controlled, and the information contained in the content, is under the control of a content provider. Unauthorized web server users must be prevented from modifying content and the risk from malicious content (as opposed to malicious users) must be minimized. A complete description of the TOE may be found in Section 2.0 of this PP.

This PP defines:

- assumptions about the security aspects of the environment in which the TOE will be used;

- threats that are to be addressed by the TOE;

- security objectives of the TOE and its environment;

- functional and assurance requirements to meet those security objectives; and

- rationale demonstrating how the requirements meet the security objectives, and how the security objectives address the threats.

1.3. Conventions

The notation, formatting, and conventions used in this PP are largely consistent with those used in Version 2.1 of the Common Criteria (CC). Selected presentation choices are discussed here to aid the PP user.

The CC allows several operations to be performed on functional requirements; *refinement*, *selection*, *assignment*, and *iteration* are defined in paragraph 2.1.4 of Part 2 of the CC. Each of these operations is used in this PP.

The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the length of a password. The **selection** operation is used to select one or more options provided by the CC in stating a requirement. Completed assignment and selection operations are denoted by *italicized text*.

Iteration of a component is required when an operation within the component must be completed multiple times with differing values, or for different allocation of functions to partitions of the TOE. Iterated functional and/or assurance components are given unique identifiers by appending a slash ("/") and an iteration identifier to the element identifiers from the CC. (e.g. FDP_ACF.1.1/CP, FDP_ACF.1.2/CP)

The **refinement** operation is used to provide an elaboration of an existing CC element to explicitly meet stated objectives. Refinement of elements is denoted by **bold** text.

Application notes document guidance for how the requirement to be applied. Rather than being collected into a separate section, the application notes are integrated with requirements and indicated as notes. Application notes should be considered informative.

In the requirement sections, each section that represents a requirement family or component, there is a mnemonic in parenthesis. These refer to the requirement section in the CC from which it was derived. Requirement elements have these references includes as superscripted text at the end of the element.

1.4. Related Protection Profiles

The protection profiles related to this PP fall into three categories:

- **Interfacing Protection Profiles**. These PPs define the security requirements for applications that interface with the Web Server. This includes the Web Browser Protection Profile, which provides the security requirements to support the end-user interface to the web server, as well as a Web Application Protection Profile, which define the security requirements for executable web content.

- **Application Protection Profiles**. These PPs define the security requirements for other networking applications that can directly or indirectly interface with the Web Server, such as servers for other Internet protocols.

- **Platform Protection Profiles**. These PPs define appropriate security requirements for underlying platforms. This includes the Controlled Access Protection Profiles (CAPP), as well as other Operating System Protection Profiles that provide basic or stronger robustness.

1.5. Protection Profile Organization

Section 1.0, INTRODUCTION, provides document management and overview information necessary to identify the PP along with references to other related PP's.

Section 2.0, TOE DESCRIPTION, defines the TOE and establishes the context of the TOE by referencing generalized security requirements.

Section 3.0, TOE ENVIRONMENT, describes the expected environment in which the TOE is to be used. This section defines the set of threats that are relevant to the secure operation of the TOE, organizational security policies with which the TOE must comply, and secure usage assumptions applicable to this analysis.

Section 4.0, SECURITY OBJECTIVES, defines the set of security objectives to be satisfied by the TOE and by the TOE operating environment.

Section 5.0, IT SECURITY REQUIREMENTS, defines the security functional and assurance requirements derived from the Common Criteria, Part 2 and Part 3, respectively, that must be satisfied by the TOE and the Non-IT environment.

Section 6.0, RATIONALE, provides rationale to demonstrate that the security objectives satisfy the threats and policies. This section also explains how the set of requirements are complete relative to the security objectives and presents a set of arguments that address dependency analysis and Strength of Function (SOF) and use of the explicit requirement.

Section 7.0, REFERENCES, provides background material for further investigation by users of the PP.

Section 8.0, TERMINOLOGY, provides a listing of definitions of terms.

Section 9.0, ACRONYMS, provides a listing of acronyms used throughout the document.

2.0. TOE DESCRIPTION

2.1. Overview of the TOE

The "World Wide Web" (colloquially called the "web") is a system for exchanging information over the Internet. It was originally designed to receive anonymous requests from unauthenticated hosts on the Internet, and to deliver the requested information in a quick and efficient manner. Today, many organizations use the web for distributing restricted-access documents within their own organizations and between organizations and individuals.

In some senses, however, the picture of a unified Web interface is a falsehood. The web is actually a collection of Internet protocols, all invoked through a common client interface (the "browser") and referenced by a Universal Resource Locator (URL). Each URL specifies the protocol, the network host serving the protocol, and the source of the content.

Many protocols are usable in URLs: HTTP, FTP, Gopher, News protocols, etc. Web servers only handle the HTTP protocols, and provide support for encryption through the SSL and TSL protocols. Other protocols are redirected by the browser to the protocol-specific servers that handle them. The security of non-HTTP protocols are not directly addressed by this profile; instead, they are addressed by specific protection profiles for each server. However, when constructing a real system, a holistic approach must be taken, and the assumptions and IT environment requirements of all protocol servers in use must be verified and reconciled.

Web servers are application programs. They execute on a host platform that provides the underlying abstractions used to store content and execute programs. The web server controls access to information by the use of its own security features in combination with the features provided by the host platform.

The main data object handled by a web server is **content**. Content represents information provided by a **content provider** to a web user. This information may be static (i.e., a pure HTML page), or dynamic (i.e., generated on the fly, either being assembled by the server or as the output of executable content). Some content is **public content**, which means that it is available to any web user that requests it without authentication. Other content is **controlled-access content**, which means that the content is distributed only to web users authorized for that content by the content provider. Note that each content provider has control over the sets of web users authorized to access their content.

Overseeing everything is the **web administrator**. The web administrator establishes the configuration of the server, and controls the set of authorized content providers.

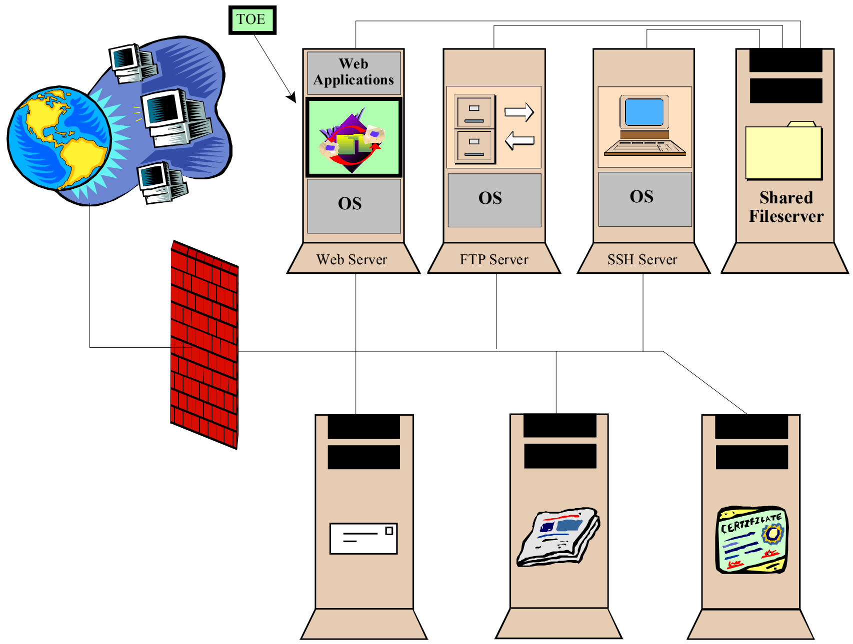

Figure 2-1. Placement of the Web Server TOE in an Overall System Architecture

In this PP, the TOE is a web server supporting SSL v3.0 or TSL. This protocol support implies support for personal digital certificates. It provides support for the serving of public content through either unencrypted (HTTP) or encrypted (HTTPS) protocols. It provides support for the serving of controlled-access content through encrypted protocols that utilize certificates.[1] Identification of the individual web user to the content provider is achievable through multiple mechanisms, such as personal certificates that are specific to the individual, or through user ID and password schemes. It is possible that the user ID and password scheme may be implemented through web application code (as opposed to the server itself), with the application enforcing its own access control. Security controls implemented by the web applications are beyond the scope of this profile, and are addressed by the Web Application Protection Profiles, unless the web applications are included as part of the TOE.

The TOE has the capability to interface with other systems, such as Directory Services, for authentication and PKI certificate storage. It may also share file systems on the underlying platform with other network services. Figure 2-1 provides the conceptual model of the TOE's placement in an overall network. This picture shows the web server sharing a file system with an FTP server and a terminal server. In such a situation, the content providers would use the FTP server and terminal servers to update content. The web server may provide a remote access

[1] It may also provide support for password protection and the serving of password protected content over unencrypted connections, but such support is not a secure usage for protected data, and is assumed not to be used by those who consider their data controlled access.

terminal interface as well, but this interface would be restricted to web server administration. Alternately, multiple forms of network application service (web server, FTP server, terminal server) could be located on the same machine. The key points, applicable to both configurations, is that the operating systems provide low-level mediation of access to files, based on a set of users common to the application servers.

The web server administrator and the web user will be required to authenticate themselves to the web server in order to access any non-public web server resources and perform their respective functions based upon their specific roles. Content providers are also required to authenticate themselves in order to access and modify their content. Any error detected during the authentication process will be logged for further investigation.

The TOE will implement certain cryptographic protocols so that information is restricted from public access. These cryptographic protocols will allow the client and server resources to exchange information in a secure manner.

In an effort to secure the data stored within the TOE, the TOE administrator and the content provider have the capability to control the access of the authenticated and authorized web user. The TOE administrator has the authority to assign various levels of access to the content providers and web users in order to prevent unauthorized access to data, modification of data, or uploading malicious code to the TOE which could result in corruption of the web server resources and/or denial of service attacks for the web users. The content provider has the ability to further refine that access.

2.2. Selection of Robustness Level

2.2.1. TOE Environment Defining Factors

In trying to specify the environments in which TOEs with various levels of robustness are appropriate, it is useful to first discuss the two defining factors that characterize that environment: **value of the resources** and **authorization of the entities** to those resources.

In general terms, the environment for a TOE can be characterized by the authorization (or lack of authorization) the least trustworthy entity has with respect to the highest value of TOE resources (i.e. the TOE itself and all of the data processed by the TOE).

Note that there are an infinite number of combinations of entity authorization and value of resources; this conceptually "makes sense" because there are an infinite number of potential environments, depending on how the resources are valued by the organization, and the variety of authorizations the organization defines for the associated entities. These two environmental factors will be related to the robustness required for selection of an appropriate TOE.

2.2.1.1. Value Of Resources

Value of the resources associated with the TOE includes the data being processed or used by the TOE, as well as the TOE itself (for example, a real-time control processor). "Value" is assigned by the using organization. For example, in the DoD low-value data might be equivalent to data marked "FOUO", while high-value data may be those classified Top Secret. In a commercial enterprise, low-value data might be the internal organizational structure as captured in the corporate on-line phone book, while high-value data might be corporate research results for the next generation product.

2.2.1.2. Authorization of Entities

Authorization that entities (users, administrators, other IT systems) have with respect to the TOE (and thus the resources of that TOE, including the TOE itself) is an abstract concept reflecting a combination of the trustworthiness of an entity and the access and privileges granted to that entity with respect to the resources of the TOE. For instance, entities that have total authorization to all data on the TOE are at one end of this spectrum; these entities may have privileges that allow them to read, write, and modify anything on the TOE, including all TSF data. Entities at the other end of the spectrum are those that are authorized to few or no TOE resources. For example, in the case of a router, non-administrative entities may have their packets routed by the TOE, but that is the extent of their authorization to the TOE's resources. For example, an entity may not be allowed to log on to the TOE at all (that is, they are not valid users listed in the TOE user database).

It is important to note that authorization **does not** refer to the **access** that the entities actually have to the TOE or its data. For example, suppose the owner of the system determines that no one other than employees was authorized to certain data on a TOE, yet they connect the TOE to the Internet. There are millions of entities that are not **authorized** to the data (because they are not employees), but they actually have connectivity to the TOE through the Internet and thus can attempt to access the TOE and its associated resources.

Entities are characterized according to the value of resources to which they are authorized; the extent of their authorization is implicitly a measure of how trustworthy the entity is with respect to compromise of the data (that is, compromise of any of the applicable security policies; e.g., confidentiality, integrity, availability). In other words, in this model the greater the extent of an entity's authorization, the more trustworthy (with respect to applicable policies) that entity is.

2.2.2. Selection Of Appropriate Robustness Levels

Robustness is a characteristic of a TOE defining how well it can protect itself and its resources; a more robust TOE is better able to protect itself. This section relates the defining factors of IT environments, authorization, and value of resources to the selection of appropriate robustness levels.

When assessing any environment with respect to Information Assurance the critical point to consider is the likelihood of an attempted security policy compromise, which was characterized in the previous section in terms of entity authorization and resource value. As previously mentioned, robustness is a characteristic of a TOE that reflects the extent to which a TOE can protect itself and its resources. It follows that as the likelihood of an attempted resource compromise increases, the robustness of an appropriate TOE should also increase.

It is critical to note that several combinations of the environmental factors will result in environments in which the likelihood of an attempted security policy compromise is similar. Consider the following two cases:

The first case is a TOE that processes only low-value data. Although the organization has stated that only its employees are authorized to log on to the system and access the data, the system is connected to the Internet to allow authorized employees to access the system from home. In this case, the least trusted entities would be unauthorized entities (e.g. non-employees) exposed to the TOE because of the Internet connectivity. However, since only low-value data are being processed, the likelihood that unauthorized entities would find it worth their while to attempt to compromise the data on the system is low and selection of a basic robustness TOE would be appropriate.

The second case is a TOE that processes high-value (e.g., classified) information. The organization requires that the TOE be stand-alone, and that every user with physical and logical access to the TOE undergo an investigation so that they are authorized to the highest value data on the TOE. Because of the extensive checks done during this investigation, the organization is assured that only highly trusted users are authorized to use the TOE. In this case, even though high value information is being processed, it is unlikely that a compromise of that data will be attempted because of the authorization and trustworthiness of the users and once again, selection of a basic robustness TOE would be appropriate.

The preceding examples demonstrate that it is possible for radically different combinations of entity authorization/resource values to result in a similar likelihood of an attempted compromise. As mentioned earlier, the robustness of a system is an indication of the protection being provided to counter compromise attempts. Therefore, a basic robustness system should be sufficient to counter compromise attempts where the likelihood of an attempted compromise is low. The following chart depicts the "universe" of environments characterized by the two factors discussed in the previous section: on one axis is the authorization defined for the least trustworthy entity, and on the other axis is the highest value of resources associated with the TOE.

As depicted in the following figure, the robustness of the TOEs required in each environment steadily increases as one goes from the upper left of the chart to the lower right; this corresponds to the need to counter increasingly likely attack attempts by the least trustworthy entities in the environment. Note that the shading of the chart is intended to reflects the notion that different environments engender similar levels of "likelihood of attempted compromise", signified by a similar color. Further, the delineation between such environments is not stark, but rather are finely grained and gradual.

While it would be possible to create many different "levels of robustness" at small intervals along the "Increasing Robustness Requirements" line to counter the increasing likelihood of attempted compromise due to those attacks, it would not be practical nor particularly useful. Instead, in order to implement the robustness strategy where there are only three robustness levels: Basic, Medium, and High, the graph is divided into three sections, with each section corresponding to set of environments where the likelihood of attempted compromise is roughly similar. This is graphically depicted in Figure 2-2.

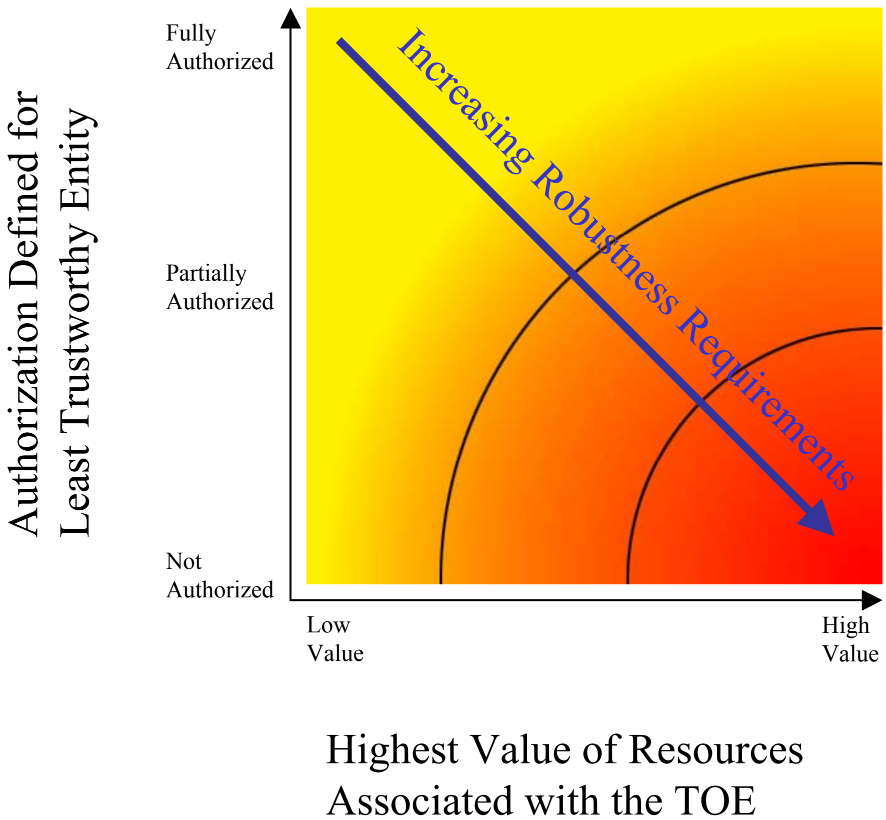

Figure 2-2. Robustness Related to Authorization and Resource Value

In the representation of environments and the robustness plane in Figure 2-3 below, the "dots" represent given instantiations of environments; like-colored dots define environments with a similar likelihood of attempted compromise. Correspondingly, a TOE with a given robustness should provide sufficient protection for environments characterized by like-colored dots. In choosing the appropriateness of a given robustness level TOE PP for an environment, then, the user must first consider the lowest authorization for an entity as well as the highest value of the resources in that environment. This should result in a "point" in the chart above, corresponding to the likelihood that that entity will attempt to compromise the most valuable resource in the environment. The appropriate robustness level for the specified TOE to counter this likelihood can then be chosen.

The difficult part of this activity is differentiating the authorization of various entities, as well as determining the relative values of resources; (e.g., what constitutes "low value" data vs. "medium value" data). Because every organization will be different, a rigorous definition is not possible. In Section 3.1.2 of this PP, the targeted threat level for a basic robustness TOE is characterized. This information is provided to help organizations using this PP insure that the functional requirements specified by this basic robustness PP are appropriate for their intended application of a compliant TOE.

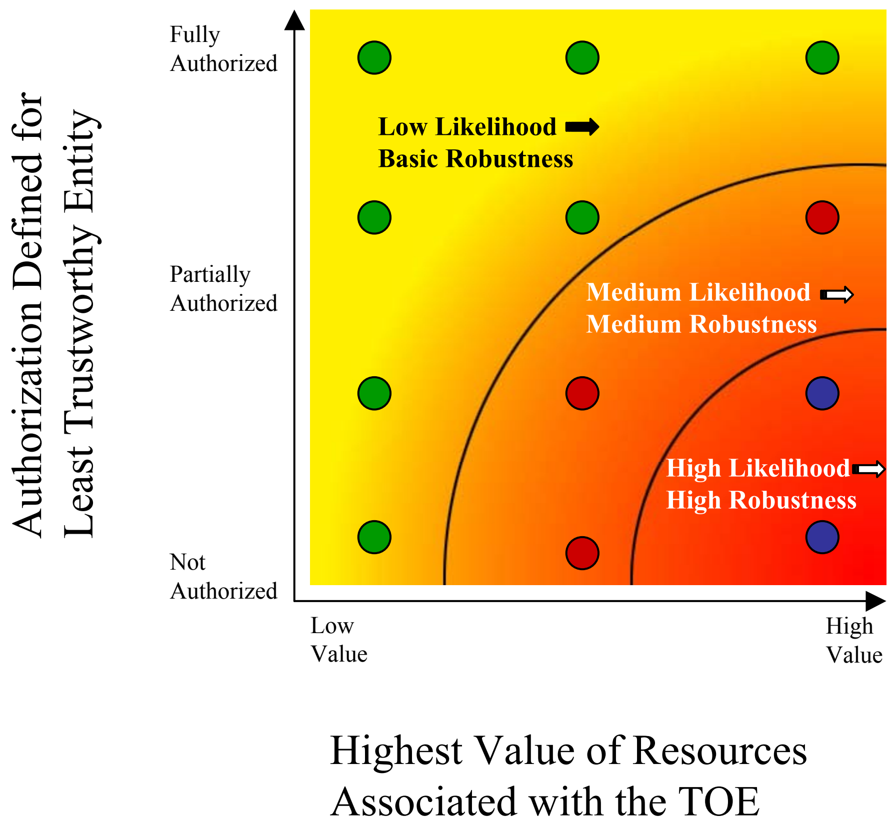

Figure 2-3. Selecting Appropriate Robustness for Environments

Basic robustness TOEs fall in the upper left area of the previously discussed robustness figures. A Basic Robustness TOE is considered sufficient for low threat environments or where compromise of protected information will not have a significant impact on mission objectives. This implies that the motivation of the threat agents will be low in environments that are suitable for TOEs of this robustness. In general, basic robustness results in "good commercial practices" that counter threats based in casual and accidental disclosure or compromise of data protected by the TOE.

Threat agent motivation can be considered in a variety of ways. One possibility is that the value of the data process or protected by the TOE will generally be seen as of little value to the adversary (i.e., compromise will have little or no impact on mission objectives). Another possibility, (where higher value data is processed or protected by the TOE) is that procuring organizations will provide other controls or safeguards (i.e., controls that the TOE itself does not enforce) in the fielded system in order to increase the threat agent motivation level for compromise beyond a level of what is considered reasonable or expected to be applied.

3.0. TOE ENVIRONMENT

3.1. Secure Usage Assumptions

This sub-section describes security aspects of the environment in which the TOE will be used or is intended to be used. This includes information about the physical, personnel, and connectivity aspects of the environment.

3.1.1. Basic Robustness PP Common Assumptions

The following assumptions are common across all Protection Profiles at the Basic Robustness level. After the general statement of the assumption, specifics are given about the assumption in the context of the Web Server Protection Profile.

A.NO_GENERAL_PURPOSE There are no general purpose computing or storage repository capabilities (e.g., compilers, editors, or user applications) available on the TOE**, except those permitted by the web administrator**.

For the Web Server Protection Profile, this means that the web server itself does not provide any such applications. Additionally, it means that the environment in which the Web Server is used does not supply any such tools. These tools may be provided on a distinct machine that shares a file system with the Web Server, but such tools are not available through the web server interface.

The only exception to this is programs required to execute approved web application executable content; these programs must be explicitly permitted by the web administrator for use through the web server interface.

A.NO_EVIL Administrators are non-hostile, appropriately trained and follow all administrator guidance.

This assumption applies without clarification to the Web Server Protection Profile.

A.PHYSICAL Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the IT environment.

The physical security described here applies only to the web server and the underlying platform. It does not apply to network lines or the browsers being used by the end users.

3.1.2. Web Server Protection Profile Assumptions

The following assumptions are specific to the context of the web server:

A.PROVIDERS_GOOD	Content providers will appropriate control the visibility of their content (i.e., they will establish appropriate access controls) based on the sensitivity of that content. Content providers will also follow published guidance regarding the installation of content.
A.SYSTEM_HIGH	All web users who can access the system have legal authorization for the information, although they may not have need to know. *This assumption means that discretionary access controls are sufficient; controls based on information labels are not necessary.*

3.2. Threats

3.2.1. Threat Agent Characterization

In addition to helping define the robustness appropriate for a given environment, the threat agent is a key component of the formal threat statements in the PP. Threat agents are typically characterized by a number of factors such as *expertise*, *available resources*, and *motivation*. Because each robustness level is associated with a variety of environments, there are corresponding varieties of specific threat agents (that is, the threat agents will have different combinations of motivation, expertise, and available resources) that are valid for a given level of robustness. The following discussion explores the impact of each of the threat agent factors on the ability of the TOE to protect itself (that is, the robustness required of the TOE).

The *motivation* of the threat agent seems to be the primary factor of the three characteristics of threat agents outlined above. Given the same expertise and set of resources, an attacker with low motivation may not be as likely to attempt to compromise the TOE. For example, an entity with no authorization to low value data none-the-less has low motivation to compromise the data; thus a basic robustness TOE should offer sufficient protection. Likewise, the fully authorized user with access to highly valued data similarly has low motivation to attempt to compromise the data, thus again a basic robustness TOE should be sufficient.

Unlike the motivation factor, however, the same can't be said for *expertise*. A threat agent with low motivation and low expertise is just as unlikely to attempt to compromise a TOE as an attacker with low motivation and high expertise; this is because the attacker with high expertise does not have the motivation to compromise the TOE even though they may have the expertise to do so. The same argument can be made for *resources* as well.

Therefore, when assessing the robustness needed for a TOE, the motivation of threat agents should be considered a "high water mark". *That is, the robustness of the TOE should increase as the motivation of the threat agents increases.*

Having said that, the relationship between expertise and resources is somewhat more complicated. In general, if resources include factors other than just raw processing power (money, for example), then expertise should be considered to be at the same "level" (low, medium, high, for example) as the resources because money can be used to purchase expertise. Expertise in some ways is different, because expertise in and of itself does not automatically procure resources. However, it may be plausible that someone with high expertise can procure the requisite amount of resources by virtue of that expertise (for example, hacking into a bank to obtain money in order to obtain other resources).

It may not make sense to distinguish between these two factors; in general, it appears that the only effect these may have is to lower the robustness requirements. For instance, suppose an organization determines that, because of the value of the resources processed by the TOE and the trustworthiness of the entities that can access the TOE, the motivation of those entities would be "medium". This normally indicates that a medium robustness TOE would be required because the likelihood that those entities would attempt to compromise the TOE to get at those resources is in the "medium" range. However, now suppose the organization determines that the entities (threat agents) that are the least trustworthy have no resources and are unsophisticated. In this case, even though those threat agents have medium motivation, the likelihood that they would be able to mount a successful attack on the TOE would be low, and so a basic robustness TOE may be sufficient to counter that threat.

It should be clear from this discussion that there is no "cookbook" or mathematical answer to the question of how to specify exactly the level of motivation, the amount of resources, and the degree of expertise for a threat agent so that the robustness level of TOEs facing those threat agents can be rigorously determined. However, an organization can look at combinations of these factors and obtain a good understanding of the likelihood of a successful attack being attempted against the TOE. Each organization wishing to procure a TOE must look at the threat factors applicable to their environment; discuss the issues raised in the previous paragraph; consult with appropriate accreditation authorities for input; and document their decision regarding likely threat agents in their environment.

The important general points we can make are:

1) The motivation for the threat agent defines the upper bound with respect to the level of robustness required for the TOE

2) A threat agent's expertise and/or resources that is "lower" than the threat agent's motivation (e.g., a threat agent with high motivation but little expertise and few resources) may lessen the robustness requirements for the TOE (see next point, however).

3) The availability of attacks associated with high expertise and/or high availability of resources (for example, via the Internet or "hacker chat rooms") introduces a problem when trying to define the expertise of, or resources available to, a threat agent.

3.2.2. Threats Addressed by the TOE

The following threats are addressed by the TOE:

T.ACCIDENTAL_ADMIN_ ERROR	An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms. *This threat exists for a web server. The server configuration is extremely important, for it determines what hosts can connect with the web server, on what ports, using what protocols, and what actions are permitted.*
T.ACCIDENTAL_AUDIT_ COMPROMISE	A user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's action. *Audit, from the point of view of a web server, are the records of pages served and modifications made through the web server.*
T.MASQUERADE	A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources. *In the web server context, this refers to the presentation of fake credentials. Note that certain aspects of addressing this threat are more in the scope of a web application, for it is the application that must determine the validity of cookies received from a browser.*
T.POOR_DESIGN	Unintentional errors in requirements specification or design of the TOE may occur, leading to flaws that may be exploited by a casually mischievous user or program.
T.POOR_IMPLEMENTATION	Unintentional errors in implementation of the TOE design may occur, leading to flaws that may be exploited by a casually mischievous user or program.
T.POOR_TEST	Lack of or insufficient tests to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may result in incorrect TOE behavior being undiscovered thereby causing potential security vulnerabilities.

T.RESIDUAL_DATA	A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another. *Web servers tend not to create their own objects, rather implying additional meaning or controls to objects managed by the IT environment. However, internal structures are used when pages get allocated for transmission or protocols negotiated, and this threat addresses the reuse of those structures.*
T.TSF_COMPROMISE	A user or process may cause, through an unsophisticated attack, TSF data, or executable code to be inappropriately accessed (viewed, modified, or deleted). *In a web server context, this refers to protection of the underlying code of the web server, the protection of configuration information managed by both the web server and content providers, as well as the protection of content that has been marked executable, even though it is not the TSF that actually executes the content.*
T.UNATTENDED_ SESSION	A user may gain unauthorized access to an unattended session. *This threat is not applicable for web users, as HTTP sessions are not connection oriented. However, it is applicable for remote sessions used for web administration.*
T.UNAUTHORIZED_ ACCESS	A user may gain access to user data for which they are not authorized according to the TOE security policy. *This threat is straightforward; in a web server context, it refers to the serving of pages for which the web users has not been authorized.*
T.UNIDENTIFIED_ACTIONS	The administrator may not have the ability to notice potential security violations, thus limiting the administrator's ability to identify and take action against a possible security breach.

T.SERVER_DISRUPT	Disruption of power, interface failures, or software failures that result in the TOE faulting to an unsecured state.

3.2.3. Threats addressed by the IT Environment

The following threats are addressed by the IT environment.

T.REPLAY	A threat agent may replay valid identification and authentication information that has been captured to disguise itself as an Authorized Administrator of the TOE.
T.TCPIP_ATTACK	A threat agent may take advantage of a published vulnerability against protocols layered below HTTP (e.g., TCP or IP), resulting in the TOE being unable to respond properly to valid requests.
T.UNDERLYING_PROT	A threat agent may be able to obtain unauthorized access to TSF data or contents through inadequate handling of TOE requests to protect underlying data objects.
T.VIRTUAL_ADDR_FAILURE	A threat agent may be able to subvert the TOE through the execution of another process on the IT platform, which modifies the operational code or data of the TOE.

3.3. Organizational Security Policies

PP-compliant TOEs must address the organizational security policies described below.

P.ACCESS_BANNER	The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the system. *TOE Access Banners apply only to remote access for administration of the web server. Access banners that would be displayed on the accessing of content are the responsibility of the content provider. Access banners that would be displayed upon local access of the web server are not applicable, for the web server is an application on the IT platform, and it is the platform itself that would be responsible for the access banners.*

P.ACCOUNTABILITY	The authorized users of the TOE shall be held accountable for their actions within the TOE.
P.CRYPTOGRAPHIC_FUNCTIONS	The TOE shall provide cryptographic functions for its own use, including encryption/decryption and digital signature operations.
P.CRYPTOGRAPHY_VALIDATED	Where the TOE requires NIST-approved security functions, only NIST FIPS validated cryptography (methods and implementations) are acceptable for key management (i.e.; generation, access, distribution, destruction, handling, and storage of keys) and cryptographic services (i.e.; encryption, decryption, signature, hashing, key distribution, and random number generation services).
P.RATINGS_MAINTENANCE	Procedures to maintain the TOE's rating must be in place, and these procedures must be implemented to maintain the TOE's rating once it is evaluated.

4.0. SECURITY OBJECTIVES

This chapter describes the security objectives for the TOE and the TOE's operating environment. The security objectives are divided between TOE Security Objectives (i.e., security objectives addressed directly by the TOE) and Security Objectives for the Operating Environment (i.e., security objectives addressed by the IT domain or by non-technical or procedural means).

4.1. TOE Security Objectives

This section defines the security objectives that are to be addressed by the TOE.

O.ADMIN_GUIDANCE	The TOE will provide administrators with the necessary information for secure management.
O.AUDIT_GENERATION	The TOE will provide the capability to detect and create records of security-relevant events associated with users.
O.AUDIT_PROTECTION	The TOE will provide the capability to protect audit information.
O.AUDIT_REVIEW	The TOE will provide the capability to selectively view audit information.
O.CONFIGURATION_IDENTIFICATION	The configuration of the TOE will be fully identified in a manner that will allow implementation errors to be identified, corrected with the TOE being redistributed promptly.
O.CORRECT_ TSF_OPERATION	The TOE will provide the capability to test the TSF to ensure the correct operation of the TSF at a customer's site.
O.CRYPTOGRAPHY_VALIDATED	The TOE will use NIST FIPS 140-2 validated cryptomodules for cryptographic services implementing NIST-approved security functions and random number generation services used by cryptographic functions.
O.CRYPTOGRAPHIC_FUNCTIONS	The TOE will provide cryptographic functions for its own use, including encryption/decryption and digital signature operations.
O.DISPLAY_BANNER	The TOE will display an advisory warning regarding use of the TOE by remote administrators.

O.DOCUMENTED_DESIGN	The design of the TOE will be adequately and accurately documented.
O.MANAGE	The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.
O.MEDIATE	The TOE will protect content provider data in accordance with its security policy.
O.PARTIAL_FUNCTIONAL_TESTING	The TOE will undergo some security functional testing that demonstrates the TSF satisfies some of its security functional requirements.
O.RATINGS_MAINTENANCE	Procedures to maintain the TOE's rating will be documented and followed.
O.RESIDUAL_ INFORMATION	The TOE will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated.
O.PARTIAL_SELF_PROTECTION	The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure through its own interfaces.
O.SAFE_RECOVERY	The TSF will provide the ability to recover to a secure state.
O.TIME_STAMPS	The TOE will provide reliable time stamps for accountability and protocol purposes.
O.TOE_ACCESS	The TOE will provide mechanisms that control a user's logical access to the TOE.
O.USER_CONFIDENCE	The TOE will provide mechanisms that permit web users to have confidence that received controlled-access data comes from the TOE.
O.VULNERABILITY_ANALYSIS	The TOE will undergo some vulnerability analysis demonstrate the design and implementation of the TOE does not contain any obvious flaws.

4.2. Security Objectives for the Operating Environment

This section defines the security objectives that are to be addressed by the IT domain or by non-technical or procedural means. The mapping and rationale for the security objectives are described in Section 6.

OE.NO_GENERAL_PURPOSE	The IT environment will provide no general-purpose compilers or interpreted except those explicitly needed to support executable content authorized by the web administrator.
OE.PROVIDERS_GOOD	Sites using the TOE will provide content providers with guidance on how to protect controlled access information and how to develop safe and appropriate content.
OE.SYSTEM_HIGH	Sites using the TOE will ensure that all authorized users of and networks connecting to the TOE have a legal ability to see the information provided (even if they may lack need to know).
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, will be provided by the IT environment.
OE.NO_EVIL	Sites using the TOE will ensure that administrators are non-hostile, appropriately trained and follow all administrator guidance.
OE.AS_REMOTE_ACCESS	A protected communication path will be provided to Administrators of the web server to permit direct (e.g., console port) or remote (e.g., physically or through encryption) administration.
OE.SEP_ENVIRONMENT	The IT Environment will provide sufficient mechanisms to protect the TSF's data and memory during storage and execution.
OE.RELIABLE_TIME_STAMP	The IT Environment will provide reliable time stamps.
OE.ACCESS_CONTROL	The IT Environment will provide the TOE with an access control mechanism suitable to protect TSF and content provider data and configuration.

OE.BASIC_ROBUSTNESS	The IT Environment will be sufficient robust to protect against the casual attacker using published exploits.
OE.AUTHORIZED_USERS	The IT Environment will ensure that all users using the IT Environment directly have been authorized and are accountable for their actions.

5.0. IT SECURITY REQUIREMENTS

This section provides functional and assurance requirements that must be satisfied by PP-compliant web server. These requirements consist of functional components from Part 2 of the CC and an Evaluation Assurance Level (EAL) containing assurance components from Part 3.

5.1. TOE Functional Security Requirements

The functional security requirements for the TOE consist of the following components derived from Part 2 of the CC, summarized in Table 5-1, below, together with the explicitly specified requirements in Table 5-2, below.

Table 5-1. Security Functional Requirements

Functional Components (from CC Part 2)	
FAU_GEN.1-NIAP-0410	Audit Data Generation
FAU_GEN.2-NIAP-0410	User Identity Association
FAU_SAR.1	Audit Review
FAU_SAR.2	Restricted audit review
FAU_SAR.3	Selectable Audit Review
FAU_SEL.1-NIAP-0407	Selectable Audit
FAU_STG.1-NIAP-0429	Protected audit trail storage
FAU_STG.NIAP-0414-1-NIAP-0429	Site-configurable Prevention of audit data loss
FAU_STG.3	Action in case of possible audit data loss
FCO_NRO.2	Enforced proof of origin
FCS_CKM.1	Cryptographic Key Generation (using Random Number Generator)
FCS_CKM.4	Cryptographic Key Destruction
FDP_ACC.2/WU	Complete Access Control (SFP: WEBUSER)
FDP_ACF.1-NIAP-0407/WU	Security Attribute Based Access Control (SFP: WEBUSER)
FDP_UCT.1/WU	Basic Data Exchange Confidentiality (SFP: WEBUSER)
FDP_UIT.1/WU	Basic Data Exchange Integrity (SFP: WEBUSER)

Functional Components (from CC Part 2)	
FDP_ACC.2/CP	Complete Access Control (SFP: CONTENT PROVIDER)
FDP_ACF.1-NIAP-0407/CP	Security Attribute Based Access Control (SFP: CONTENT PROVIDER)
FDP_RIP.2	Full Residual Information Protection
FIA_AFL.1-NIAP-0425	Authentication Failure Handling
FIA_ATD.1	User Attribute Definition
FIA_UAU.1	Timing of Authentication
FIA_UAU.7	Protected Authentication Feedback
FIA_UID.1	Timing of Identification
FIA_USB.1-NIAP-0351	User-subject Binding
FMT_MOF.1	Management of Security Functions Behavior (Enable/Disable)
FMT_MSA.1	Management of security attributes
FMT_MSA.2	Secure Security Attributes
FMT_MSA.3-NIAP-0429	Static Attribute Initialization
FMT_MTD.1	Management of TSF Data
FMT_REV.1	Revocation
FMT_SMR.1	Security Roles
FPT_AMT.1	Abstract Machine Testing
FPT_RCV.2	Automated Recovery
FPT_RVM.1	Non-bypassibility of the TSP
FPT_STM.1	Reliable time stamps
FPT_TST.1/CR	TSF Testing (for cryptography)
FPT_TST.1/NC	TSF Testing (Non-Cryptographic Code)

Functional Components (from CC Part 2)	
FTA_SSL.1	TSF-Initiated Session Locking
FTA_SSL.2	User-Initiated Locking
FTA_SSL.3/IN	TSF-Initiated Termination
FTA_SSL.3/WU	Web User Termination
FTA_TAB.1	Default TOE Access Banners
FTP_ITC.1	Inter-TSF trusted channel

Table 5-2. Explicit Security Functional Requirements

Explicit Functional Components	
FCS_BCM_EXP.1	Baseline Cryptographic Module
FCS_CKM_EXP.1	Cryptographic Key Establishment
FCS_COP_EXP.1	Random number generation
FCS_COP.1 (2)	Cryptographic Operation (Digital Signature Generation/Verification)
FCS_COP.1 (3)	Cryptographic Operation (Cryptographic Hashing Function)
FPT_SEP_EXP.1	TSF Domain Separation; protect from interference
FPT_TST_EXP.1/KG	TSF Testing (Key Generation Code)

5.1.1. FAU: Security Audit

5.1.1.1. FAU_GEN.1-NIAP-0410: Audit data generation

5.1.1.1.1. The TSF shall be able to generate an audit record of the following auditable events: (FAU_GEN.1.1-NIAP-0410)

(a) Start-up and shutdown of the audit functions;

(b) All auditable events listed **in Table 5-3**;

(c) [selection: [assignment: *events at a basic level of audit introduced by the inclusion of additional SFRs determined by the ST author*], [assignment: *events* *commensurate with a basic level of audit introduced by the inclusion of explicit requirements determined by the ST author*], "*no additional events*"].

Application Notes:

For the selection, the ST author should choose one or both of the assignments (as detailed in the following paragraphs), or select "no additional events".

For the first assignment, the ST author should augment the table (or lists explicitly) the audit events associated with the basic level of audit for any SFRs that the ST author includes that are not included in this PP.

For the second assignment the ST author should include audit events that may arise due to the inclusion of any explicit requirements not already in the PP. Because "basic" audit is not defined for such requirements, the ST author will need to determine a set of events that are commensurate with the type of information that is captured at the basic level for similar requirements.

If no additional (CC or explicit) SFRs are included, or if additional SFRs are included that do not have "basic" audit associated with them, then it is acceptable to assign "no additional events" in this item.

Operation Notes:

This was refined to move the list of audit events into a separate table.

5.1.1.1.2. The TSF shall record within each audit record at least the following information: (FAU_GEN.1.2-NIAP-0410)

a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, *the information specified in column three of Table 5-3*.

Application Note:

In column 3 of Table 5-3, "if applicable" is used to designate data that should be included in the audit record if it "makes sense" in the context of the event that generates the record. If no other information is required (other than that listed in "a") for a particular audit event type, then an assignment of "none" is acceptable.

Table 5-3. Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1-NIAP-0410	None	

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.2-NIAP-0410	None	
FAU_ SAR.1	None	
FAU_SAR.2	Unsuccessful attempts to read information from the audit records	The identity of the administrator performing the function
FAU_SAR.3	None	
FAU_SEL.1-NIAP-0407	All modifications to the audit configuration that occur while the audit collection functions are operating.	The identity of the administrator performing the function
FAU-STG.1-NIAP-0429	Attempts to backup or delete the audit trail.	The identity of the administrator performing the function
FAU-STG.NIAP-0414-1-NIAP-0429	None	
FAU_STG.3	Actions taken due to exceeding the audit threshold	The identity of the Security Administrator performing the function
FCO_NRO.2	The invocation of the non-repudiation service.[2]	
FCS_BCM_EXP.1	None	
FCS_CKM.1	Failure of the activity	
FCS_CKM.4	Failure of the activity	
FCS_CKM_EXP.1	Failure of the activity	
FCS_COP_EXP.1	None	
FCS_COP.1 (2)	Failure of the activity	
FCS_COP.1 (3)	None	

[2] Given the technology used, this invocation may be recorded in another fashion, such as the record of the establishment of an SSL session (as SSL provides the non-repudiation of origin).

Requirement	Auditable Events	Additional Audit Record Contents
FDP_ACC.2/WU	None	
FDP_ACF.1.1-NIAP-0407/WU	Failure of the activity	
FDP_UCT.1/WU	None	
FDP_UIT.1/WU	None	
FDP_ACC.2/CP	None	
FDP_ACF.1.1-NIAP-0407/CP	Failure of the activity	
FDP_RIP.2	None	
FIA_AFL.1-NIAP-0425	The reaching of the threshold for the unsuccessful authentication attempts and the actions (e.g. disabling of a terminal) taken and the subsequent, if appropriate, restoration to the normal state (e.g. re-enabling of a terminal).	
FIA_ATD.1	None	
FIA_UAU.1	None	
FIA_UAU.7	None	
FIA_UID.1	All use of the user identification mechanism used for authorized users (that is, those that authenticate to the TOE)	Claimed identity of the user using the identification mechanism
FIA.USB.1-NIAP-0429	None	
FMT_MOF.1	All modifications in the behavior of the functions in the TSF	The identity of the administrator performing the function
FMT_MSA.1	Modification of security attributes	The identity of the administrator performing the function
FMT_MSA.2	All manipulation of the security attributes	The identity of the administrator performing the function

Requirement	Auditable Events	Additional Audit Record Contents
FMT_MSA.3-NIAP-0429	None	
FMT_MTD.1	All modifications to the values of TSF data.	
FMT_REV.1	None	
FMT_SMR.1	None	
FMT_AMT.1	Execution of the tests of the underlying machine and the results of the tests.	
FPT_RCV.2	The fact that a failure or service discontinuity occurred. Resumption of regular operation.	
FPT_SEP_EXP.1	None	
FPT_STM.1	Changing of the time	
FPT_TST.1/CR	None	
FPT_TST.1/NC	None	
FPT_TST_EXP.1/KG	Execution of this set of TSF self tests	The identity of the administrator performing the test, if initiated by an administrator
FTA_SSL.1	Locking of an interactive session by the session locking mechanism Any attempts at unlocking of an interactive session	The identity of the user associated with the session being locked or unlocked
FTA_SSL.2	Locking of an interactive session by the session locking mechanism Any attempts at unlocking of an interactive session	The identity of the user associated with the session being locked or unlocked

Requirement	Auditable Events	Additional Audit Record Contents
FTA_SSL.3/IN	Termination of an interactive session by the session locking mechanism	The identity of the user associated with the session being locked or unlocked
FTA_SSL.3/WU	Invalidation of credential by the session locking mechanism	The identity of the user associated with the session being locked or unlocked
FTA_TAB.1	None	
FTP_ITC.1	All uses of the trusted channel functions	Identification of the initiator and the target of all trusted channel functions

-

5.1.1.2. FAU_GEN.2-NIAP-0410: User identity association

5.1.1.2.1. For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event. (FAU_GEN.2.1-NIAP-0410)

5.1.1.3. FAU_SAR.1: Audit Review

5.1.1.3.1. The TSF shall provide *the TOE administrators* with the capability to read *all information contained within the audit record* from the audit records. (FAU_SAR.1.1)

5.1.1.3.2. The TSF shall provide the audit records in a manner suitable for the user to interpret the information. (FAU_SAR.1.2)

5.1.1.4. FAU_SAR.2: Restricted Audit Review

5.1.1.4.1. The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access. (FAU_SAR.2.1)

5.1.1.5. FAU_SAR.3: Selectable Audit Review

5.1.1.5.1. The TSF shall provide the ability to perform *searches* of audit data based on **any of the following**: (FAU_SAR 3.1)

(a) *user identity;*

(b) *source subject identity;*

(c) *destination subject identity;*

(d) *ranges of one or more: dates, times, user identities, subject service identifiers, or transport layer protocol;*

(e) *TOE network interfaces; and*

(f) *[selection: [assignment: other criteria determined by the ST Author], "no additional criteria"]].*

Application Note:

It is implied that the Audit Administrator is the only user who can perform these functions, since they are the only users with read access to all of the audit records in the audit trail. Audit data should be capable of being searched and sorted on all criteria specified in a–f, if applicable (i.e., not all criteria will exist in all audit records).

Sorting means to arrange the audit records such that they are "grouped" together for administrative review. For example the Audit Administrator may want all the audit records for a specified source subject identity or range of source subject identities (e.g., IP source address or range of IP source addresses) presented together to facilitate their audit review. If no additional criteria are provided by the TOE to perform searches or sorting of audit data, the ST author selects "no additional criteria".

Operations Note:

This was refined to fix the grammatical introduction to the list.

5.1.1.6. FAU_SEL.1-NIAP-0407: Selective Audit

5.1.1.6.1. The TSF shall be able to include or exclude auditable events from the set of audited events based on the following attributes: (FAU_SEL.1.1-NIAP-0407)

(a) *user identity;*

(b) *network identifier;*

(c) *subject service identifier;*

(d) *event type;*

(e) *success of auditable security events;*

(f) *failure of auditable security events; and*

(g) *[selection: [assignment: list of additional criteria that audit selectivity is based upon], "no additional criteria"]].*

Application Note:

“user identity” applies to authenticated users; see application note for FIA_UID.2. “service identifier” is defined in FDP_IFF.1.2-NIAP-0417(*). “event type” is to be defined by the ST author; the intent is to be able to include or exclude classes of audit events.

5.1.1.7. FAU_STG.1-NIAP-0429: Protected audit trail storage

5.1.1.7.1. The TSF shall protect the stored audit records in the audit trail from unauthorized deletion. (FAU_STG.1.1-NIAP-0429)

5.1.1.7.2. The TSF shall be able to *prevent* modifications to the audit records in the audit trail. (FAU_STG.1.2-NIAP-0429)

5.1.1.8. FAU_STG.NIAP-0414-1-NIAP-0429: Site-configurable Prevention of audit data loss

5.1.1.8.1. The TSF shall provide an authorized administrator with the capability to select one or more of the following actions to be taken if the audit trail is full**:** (FAU_STG.NIAP-0414-1-NIAP-0429-1.1)

(a) *prevent auditable events, except those taken by the authorized user with special rights*

(b) *overwrite the oldest stored audit records*

(c) [selection: [assignment: other actions to be taken in case of audit storage failure], "no additional options"]

Operations Note:

This was refined to make the embedded multiple-choice selection into a list.

5.1.1.8.2. The TSF shall overwrite the oldest stored audit records if the audit trail is full and no other action has been selected. (FAU_STG.NIAP-0414-1-NIAP-0429.1.2)

Application Note:

The TOE provides the administrator the option of preventing audit data loss by preventing auditable events from occurring. The administrator’s actions under these circumstances are not required to be audited. The TOE also provides the administrator the option of overwriting “old” audit records rather than preventing auditable events, which may protect against a denial-of-service attack.

The ST writer should fill in other technology-specific actions that can be taken for audit storage failure (in addition to the two already specified), or select “no additional options” if there are no such technology-specific actions.

5.1.1.9. FAU_STG.3: Action in case of possible audit data loss

5.1.1.9.1. The TSF shall *immediately alert the administrators by displaying a message at the local console, [selection: [assignment: other actions determined by the ST author], "none"]]* if the audit trail exceeds *an Administrator-settable percentage of storage capacity*. (FAU_STG.3.1)

Application Note:

The ST Author should determine if there are other actions that should be taken when the audit trail setting is exceeded, and put these in the assignment. If there are no other actions, then the ST Author should select "none".

5.1.2. FCO: Communication

5.1.2.1. FCO_NRO.2 Enforced proof of origin

5.1.2.1.1. The TSF shall enforce the generation of evidence of origin for transmitted *controlled-access content* at all times. (FCO_NRO.2.1)

5.1.2.1.2. The TSF shall be able to relate the [assignment: list of attributes] of the originator of the information, and the [assignment: list of information fields] of the information to which the evidence applies. (FCO_NRO.2.2)

5.1.2.1.3. The TSF shall provide a capability to verify the evidence of origin of **controlled-access** information to **the** *recipient* given *no limitations on the evidence of origin*. (FCO_NRO.2.3)

Operations Note:

This element was refined to make clear the type of information for which evidence of origin must be provided.

Application Note:

The intent of the above is to capture the notion that controlled-access content should be signed with the certificate of the generating user, and the SSL/TLS connection provides the ability to verify that signature.

5.1.3. FCS: Cryptographic Support

The cryptographic requirements are structured to accommodate use of FIPS 140-2-validated cryptographic modules (also called cryptomodules) in meeting the requirements. Since the FIPS 140-2 scheme does not cover all aspects of all algorithms, a convention is needed to distinguish the cryptographic functionality that the TSF is required to provide that cannot be provided by a FIPS-validated cryptomodule from cryptographic functionality that can be provided via a FIPS-validated cryptomodule. In the following text and requirements, "cryptomodule" is used in the very specific sense that it is

- a module that is FIPS 140-2 validated (to comply with FCS_BCM_EXP below);

- a module implementing validated NIST-approved security functions; and
- a module containing cryptographic functionality available in a NIST-approved mode.

It is the intent of these requirements (and the requirements are worded such) that whenever cryptographic functionality that can be FIPS-validated is required, that functionality be implemented in a cryptomodule. This means that when key management requirements (including key generation) are present, the key management functionality must be present in the cryptomodule. As an example, cryptomodules implementing AES must generate their own key.

It is important to note to vendors and end users that any IT entity that is used to protect National Security Information, and employs cryptography as a protection mechanism, will require the TOE's key management techniques to be approved by NSA when the TOE is fielded.

5.1.3.1. FCS_BCM_EXP.1: Baseline Cryptographic Module

5.1.3.1.1. All cryptographic modules shall be FIPS PUB 140-2 validated, and perform the specified cryptographic functions in a NIST-approved mode of operation. (FCS_BCM_EXP.1.1)

5.1.3.1.2. The cryptographic module implemented shall have a minimum overall rating of FIPS PUB 140-2 Security Level 1. (FCS_BCM_EXP.1.2)

5.1.3.2. FCS_CKM.1 Cryptographic Key Generation (using Random Number Generator)

5.1.3.2.1. The cryptomodule shall generate **symmetric** cryptographic keys **using a NIST-approved Random Number Generator** for **all key sizes** that meet **one of the standards defined in Annex C to FIPS 140-2.** (FCS_CKM.1.1)

Application Note:

Annex C to FIPS 140-2 defines NIST-approved random number generation algorithms. Each of the algorithms is defined in an associated standard listed in the Annex.

5.1.3.3. FCS_CKM.4: Cryptographic Key Destruction

5.1.3.3.1. The TSF shall destroy cryptographic keys in accordance with a cryptographic key zeroization method that meets the following: (FCS_CKM.4.1)

- Key Zeroization Requirements in FIPS PUB 140-2 Key Management Security Level 1;
- Zeroization of all plaintext cryptographic keys and all other critical cryptographic security parameters shall be immediate and complete;

- For embedded cryptographic modules, the zeroization shall be executed by overwriting the key/critical cryptographic security parameter storage area three or more times with an alternating pattern;

- If the cryptographic module contains any doors or removable covers or if a maintenance access interface is defined, then the module shall contain tamper response and zeroization circuitry. The tamper response and zeroization circuitry shall immediately zeroize all plaintext secret and private keys and critical cryptographic security parameters when a door is opened, a cover is removed, or when the maintenance access interface is accessed. The tamper response and zeroization circuitry shall remain operational when plaintext secret and private cryptographic keys or CSPs are contained within the cryptographic module; and

- When transferring any key/CSP to another location, the TSF shall overwrite each intermediate storage area for private cryptographic keys, plaintext cryptographic keys, and all other critical security parameters three or more times with an alternating pattern.

Application note:

The last item applies to locations that are used when the keys/parameters are copied during processing, and not to the locations that are used for storage of the keys, which are specified in the previous two items. The temporary locations could include memory registers, physical memory locations, and even page files and memory dumps.

5.1.3.4. FCS_CKM_EXP.1 Cryptographic Key Establishment

5.1.3.4.1. The TSF shall provide the following cryptographic key establishment technique(s): [selection: (FCS_CKM_EXP.1.1)

- Cryptographic Key Establishment using Discrete Logarithm Key Agreement

Application Note:

This element of the top-level selection applies to automated key agreement schemes where an exchange occurs between the TOE and another IT entity that results in both entities having the same secret key without ever having passed that key between the two entities. This is in contrast to key transport schemes, where key is actually passed between two IT entities. This is also distinct from key loading, where the user is either directly inputting or receiving key, or an automated device (token, PC card, etc.) is inputting or receiving key.

(a) The TSF shall provide the capability to act as the initiator or responder (that is, act as Party U or Party V as defined in the standard) to agree on cryptographic keys of all sizes using the [**selection: dhStatic, dhEphem, dhOneFlow, dhHybrid1, dhHybrid2, dhHybridOneFlow, MQV1, MQV2**] key agreement scheme where domain parameter p is a prime of **[Assignment:** ***size of prime "p" in number of bits that is 1024 or greater*]** and domain parameter q is a prime of **[Assignment:** ***size of prime "q" in number of bits that is 160 or greater*],** and that conforms with ANSI X9.42-2001, Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography.

Application Note:

It should be noted that the actual key size of the symmetric key agreed to when using this scheme will be a function of the algorithm that will be using the key, as specified in FCS_COP.1 (1).

In the selection in paragraph a), one or more of the schemes should be chosen by the ST writer, based on what schemes the TOE implements. Note that the requirement is for the TSF to be able to act as either party (as detailed in the standard) for the chosen scheme(s).

The two assignments are used to specify the number of bits used for the domain parameters p and q (which are primes). The requirement above indicates that p must be a prime of at least 1024 bits, while q must be a prime of at least 160 bits. The ST writer should fill in the appropriate number of bits based on the implementation. This applies if the implementation generates its own domain parameters, or if it obtains the domain parameters in some other way (e.g., hard-coded, obtained from an outside authority).

(b) The TSF shall conform to the standard using a NIST-approved Message Authentication Code (MAC) function, a NIST-approved Random Number generation function, and a NIST-approved Hashing function.

(c) The choices and options used in conforming to the key agreement scheme(s) are as follows: **[assignment: options that the TSF implements when implementing the selected key agreement schemes, including options for any prerequisite or dependant functions (e.g., domain parameter generation and validation.]**;

In the X9.42-2001 standard there are several sections that are marked "optional", or where a choice is given. Choices are, for example, how the domain parameters are obtained (generated or obtained from some other entity). Another example is the key derivation function that is implemented. ST writers should use the assignment to provide sufficient information so that 1) it is possible to test the implementation of the function in a repeatable fashion, and 2) readers (consumers) of the ST understand exactly what is done by the key agreement schemes implemented. The ST author should ensure that all of the prerequisite options/choices, as well as choices/options in dependant functions, are covered in the assignment.

- Cryptographic Key Establishment using Elliptic Curve Key Agreement

Application Note:

This element of the top-level selection applies to automated key agreement schemes where an exchange occurs between the TOE and another IT entity that results in both entities having the same secret key without ever having passed that key between the two entities. This is in contrast to key transport schemes, where key is actually passed between two IT entities. This is also distinct from key loading, where the user is either directly inputting or receiving key, or an automated device (token, PC card, etc.) is inputting or receiving key.

(a) The TSF shall provide the capability to act as the initiator or responder (that is, act as Party U or Party V as defined in the standard) to agree on cryptographic keys of all sizes using the [**selection: Ephemeral Unified Model, 1-Pass Diffie-Hellman, Static Unified Model, Combined Unified Model with Key Confirmation, 1-Pass Unified Model, Full Unified Model, Full Unified Model with Key Confirmation, Station-to-Station, 1-Pass MQV, Full MQV, Full MQV with Key Confirmation**] key agreement scheme using Elliptic Curves with the order of the base point being a **[Assignment:** ***size of the order of the base point "n" in number of bits that is 160 or greater*****]**-bit value, and conforms to ANSI X9.63-2001, Public Key Cryptography for the Financial Services Industry: Key Agreement and Key Transport Elliptic Curve Cryptography.

Application Note:

It should be noted that the actual key size of the symmetric key agreed to when using this scheme will be a function of the algorithm that will be using the key, as specified in FCS_COP.1 (1).

In the selection in paragraph a), one or more of the schemes should be chosen by the ST writer, based on what schemes the TOE implements. Note that the requirement is for the TSF to be able to act as either party (as detailed in the standard) for the chosen scheme(s) where the schemes are asymmetric.

The assignment is used to specify the number of bits used for the domain parameter n, which is the order of the base point of the curve chosen (the standard uses "n" to denote this value). The requirement above indicates that n must be at least a 160-bit value. The ST writer should fill in the appropriate number of bits based on the implementation. This applies if the implementation generates its own domain parameters, or if it obtains the domain parameters in some other way (e.g., hard-coded, obtained from an outside authority).

(b) The TSF shall conform to the standard using a NIST-approved MAC function, a NIST-approved Random Number generation function, and a NIST-approved Hashing function.

(c) The choices and options used in conforming to the key transport scheme(s) are as follows: **[assignment: options that the TSF implements when implementing the selected key transport schemes, including options for any prerequisite or dependant functions (e.g., domain parameter generation and validation.]**;

Application Note: In the X9.63-2001 standard there are several sections that are marked "optional", or where a choice is given. Choices are, for example, in the domain parameter generation and validation section (Section 5.1) where domain parameters can be generated over F_p or over F_2^m. Another example is the Diffie-Hellman primitive (Standard or Modified) that is implemented. ST writers should use the assignment to provide sufficient information so that 1) it is possible to test the implementation of the function in a repeatable fashion, and 2) readers (consumers) of the ST understand exactly what is done by the key agreement schemes implemented. The ST author should ensure that all of the prerequisite options/choices, as well as choices/options in dependant functions, are covered in the assignment.

- Cryptographic Key Establishment using Key Transport

Application Note:

This element of the top-level selection applies to automated key transport schemes where key is exchanged between the TOE and another IT entity. This is in contrast to key agreement schemes, where key is determined based on shared public information between two IT entities. This is also distinct from key loading, where the user is either directly inputting or receiving key, or an automated device (token, PC card, etc.) is inputting or receiving key.

(a) The TSF shall provide (act as the initiator) and accept (act as the responder) cryptographic keys to/from another IT Entity using the [**selection: 1-Pass Transport Scheme; 3-Pass Transport Scheme; both the 1-Pass and 3-Pass Transport Schemes**] using Elliptic Curves with the order of the base point being a **[Assignment:** ***size of modulus "n" in number of bits that is 160 or greater*****]**-bit value in a manner that conforms with ANSI X9.63-2001, Public Key Cryptography for the Financial Services Industry: Key Agreement and Key Transport Elliptic Curve Cryptography.

Application Note:

In the selection in paragraph a), one or more of the schemes should be chosen by the ST writer, based on what schemes the TOE implements. Note that the requirement is for the TSF to be able to act as either party (as detailed in the standard) for the chosen scheme(s).

The assignment is used to specify the number of bits used for the domain parameter n, which is the order of the base point of the curve chosen (the standard uses "n" to denote this value). The requirement above indicates that n must be at least a 160-bit value. The ST writer should fill in the appropriate number of bits based on the implementation. This applies if the implementation generates its own domain parameters, or if it obtains the domain parameters in some other way (e.g., hard-coded, obtained from an outside authority).

(b) The TSF shall conform to the standard using a NIST-approved MAC function, a NIST-approved Random Number generation function, and a NIST-approved Hashing function.

(c) The choices and options used in conforming to the key transport scheme(s) are as follows: **[assignment: options that the TSF implements when implementing the selected key transport schemes, including options for any prerequisite or dependant functions (e.g., domain parameter generation and validation.];**

Application Note:

In the X9.63-2001 standard there are several sections that are marked "optional", or where a choice is given. Choices are, for example, in the domain parameter generation and validation section (Section 5.1) where domain parameters can be generated over F_p or over $F_2{}^m$. Another example is the Diffie-Hellman primitive (Standard or Modified) that is implemented. ST writers should use the assignment to provide sufficient information so that 1) it is possible to test the implementation of the function in a repeatable fashion, and 2) readers (consumers) of the ST understand exactly what is done by the key agreement schemes implemented. The ST author should ensure that all of the prerequisite options/choices, as well as choices/options in dependant functions, are covered in the assignment.

- Cryptographic Key Establishment using Manual Loading

Application Note:

This element of the top-level selection applies to the case where a human is either typing key into the TSF, or the TSF is outputting key to a display, for instance. The distinguishing feature is that the transaction is between a human and the TSF, and **not** between the TSF and another IT device or IT media.

(a) The cryptomodule shall **[selection: be able to accept as input; be able to output in the following circumstances [*assignment: circumstances under which the cryptomodule will output a key*]]** cryptographic keys in accordance with a specified manual cryptographic key distribution method using NIST-approved Key Management techniques that meets the FIPS 140-2 Key Management Security Levels 1, Key Entry and Output.;

Application Note:

The selection should be used by the ST author to indicate whether the cryptomodule is capable of accepting key, capable of outputting key, or both. In the case where the key is output, the ST author should use the assignment to detail the conditions under which key is output from the cryptomodule (for example, only during a certain type of key generation activity).

Note that this requirement mandates that cryptomodules in the TSF have the ability to perform manual key input/output, and that this capability has been through the FIPS validation process.

- Cryptographic Key Establishment using Automated Loading

  Application Note: This element of the top-level selection applies to automated key loading device. In the case where key is being transferred from the device to the TSF the key is being “input”. In the case where the key is being transferred from the TSF to the device (for instance, a CA loading a user’s private key into a token device) the key is being “output.”

(b) The cryptomodule shall **[selection: be able to accept as input; be able to output in the following circumstances [*assignment: circumstances under which the cryptomodule will output a key*]]** cryptographic keys in accordance with a specified electronic cryptographic key distribution method using NIST-approved Key Management techniques that meet the following: [

Application Note:

The selection should be used by the ST author to indicate whether the cryptomodule is capable of accepting key, capable of outputting key, or both. In the case where the key is output, the ST author should use the assignment to detail the conditions under which key is output from the cryptomodule (for example, only during a certain type of key generation activity).

1) The electronic device is directly attached by [selection: internal bus, serial port, USB port, audio device, assignment: [*other non-network physical device*]] to the TSF;

Application Note:

An example of a device attached by an internal bus would be a floppy device used for keys transported on floppy disks.

2) The TSF shall perform key error detection scheme on keys input via electronic methods using [selection: *parity check,* [assignment: *other key error detection scheme*]; and

Application Note: The ST writer should indicate what error detection scheme is employed. The requirement above refers to errors in parity or structure of the key; it does not necessarily require checks on key “goodness”, length, format, etc.

3) FIPS 140-2 Key Management Security Levels 1, Key Entry and Output.]

Application Note:

Note that this requirement mandates that cryptomodules in the TSF have the ability to perform automated key input/output, and that this capability has been through the FIPS validation process.

Application Note:

The ST author selects one or more of the identified methods (i.e., the two key agreement schemes, key transport, manual loading or automated loading) used to establish cryptographic keys in the TOE.

5.1.3.5. FCS_COP.1 (1) Cryptographic Encryption/Decryption

5.1.3.5.1. The TSF shall perform data encryption/decryption services in accordance with a **NIST-approved** cryptographic algorithm [selection: Triple Data Encryption Algorithm (TDEA), AES] **used in NIST-approved modes of operation with** cryptographic key size of 128 bits or more that meets the following: (FCS_COP.1.1(1))

a) FIPS PUB 140-2, Security Requirements for Cryptographic Modules,

   If a cryptographic module implements a bypass capability, where services are provided without cryptographic processing (e.g., transferring plaintext through the module without encryption), then

   - two independent internal actions shall be required to activate the capability to prevent the inadvertent bypass of plaintext data due to a single error (e.g., two different software or hardware flags are set, one of which may be user-initiated), and

   - the module shall show status to indicate whether

     (1) the bypass capability is not activated, and the module is exclusively providing services with cryptographic processing (e.g., plaintext data is encrypted),

     (2) the bypass capability is activated and the module is exclusively providing services without cryptographic processing (e.g., plaintext data is not encrypted), or

     (3) the bypass capability is alternately activated and deactivated and the module is providing some services with cryptographic processing and some services without cryptographic processing (e.g., for modules with multiple communication channels, plaintext data is or is not encrypted depending on each channel configuration).

b) FIPS PUB 46-3, Data Encryption Standard, and

c) ANSI X9.52-1998, Triple Data Encryption Algorithm Modes of Operation.

Application Note:

The ST author should specify the modes in which the cryptomodule operates in the TOE. Note that these modes must be available in the NIST-approved operation mode of the cryptomodule. SP 800-38A (“Recommendation for Block

Cipher Modes of Operation") specifies five confidentiality modes that are used with any approved block cipher. The modes in SP 800-38A are updated versions of the ECB, CBC, CFB, and OFB modes that are specified in FIPS Pub. 81; in addition, SP 800-38A specifies the CTR mode.

5.1.3.6. FCS_COP.1 (2) Cryptographic Operation (Digital Signature Generation/Verification)

5.1.3.6.1. The **cryptomodule** shall perform digital signature generation and verification using the NIST-approved Security Function [selection: (FCS_COP.1.1 (2))

a. Digital Signature Algorithm (DSA) with a key size (modulus) of 1024 bits or greater,

b. RSA Digital Signature Algorithm (rDSA with odd e) with a key size (modulus) of 1024 bits or greater, or

c. Elliptic Curve Digital Signature Algorithm (ECDSA) with a key size of 160 bits or greater]

- that meets the following:

a) Case: Digital Signature Algorithm

FIPS PUB 186-2, Digital Signature Standard, for signature creation and verification processing; and ANSI Standard X9.42-2001, Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography for generation of the domain parameters;

b) Case: RSA Digital Signature Algorithm (with odd e)

ANSI X 9.31-1998 (May 1998), Digital Signatures Using Reversible Public Key Cryptography For The Financial Services Industry (rDSA);

Application Note:

In the X9.31-1998 standard there are several sections that are marked "optional", or where a choice is given. For instance, the public verification exponent "e" can be fixed or randomly generated. Another instance is that the procedure in section 4.1.2.1 can be followed to generate the primes p and q, or another procedure followed as long as the primes generated meet the conditions in section 4.1.2. The goal of the assignment is to provide sufficient information such that 1) it is possible to test the implementation of the function in a repeatable fashion, and 2) readers (consumers) of the ST understand exactly what is done by the rDSA implementation. The ST author should ensure that all of the prerequisite options/choices, as well as choices/options in dependant functions, are covered in the assignment.

c) Case: Elliptic Curve Digital Signature Algorithm

ANSI X9.62-1-1998, Public Key Cryptography for the Financial Services Industry: Elliptic Curve Digital Signature Algorithm (ECDSA).

Application Note:

The requirement above indicates the number of bits used for the domain parameter n, which is the order of the base point of the curve chosen (the standard uses "n" to denote this value). That n must be at least a 160-bit value. The ST writer should fill in the appropriate number of bits based on the implementation. This applies if the implementation generates its own domain parameters, or if it obtains the domain parameters in some other way (e.g., hard-coded, obtained from an outside authority).

5.1.3.7. FCS_COP.1 (3) Cryptographic Operation (Cryptographic Hashing Function)

5.1.3.7.1. The TSF shall perform all Cryptographic Hashing Functions (used by other cryptographic functionality of the TSF) using a NIST-approved Cryptographic Hashing Function implemented in a NIST-approved cryptomodule running in a NIST-approved mode. [(FCS_COP.1.1(3))]

Application Note:

Whenever a referenced standard calls for a cryptographic hashing capability (e.g., SHA-1), this requirement specifies the subset of cryptographic hashing functions (those that are FIPS-validated) that are acceptable. Note that the hashing function does not have to be implemented in the cryptomodule that is performing the cryptographic operation. Also note that this requirement is not calling for the hashing functionality to be made generally available (e.g., to untrusted users via an API).

5.1.3.8. FCS_COP_EXP.1: Random number generation

5.1.3.8.1. The TSF shall perform all Random Number Generation used by the cryptographic functionality of the TSF using a NIST-approved Random Number Generator implemented in a NIST-approved cryptomodule running in a NIST-approved mode. [(FCS_COP_EXP.1.1)]

Application Note:

Whenever a referenced standard calls for a random number generation capability, this requirement specifies the subset of random number generators (those that are FIPS-validated) that are acceptable. Note that the RNG does not have to be implemented in the cryptomodule that is performing the cryptographic operation. Also note that this requirement is not calling for the RNG functionality to be made generally available (e.g., to untrusted users via an API).

5.1.4. FDP/WU: User Data Protection: WEBUSER (WU) Security Functional Policy

The intent of the WEBUSER SFP is to control access by entities accessing the server over the network to obtain content. Note that this SFP contains no residual information protection requirements, as the subjects under the policy (web users) lack the ability to create, modify, or delete the objects under the policy (content).

5.1.4.1. FDP_ACC.2/WU: Complete Access Control (SFP: WEBUSER)

5.1.4.1.1. The TSF shall enforce the *WEBUSER SFP* on **the following subjects and objects,** and **upon** all operations among subjects and objects covered by **this** Security Function Policy (SFP): (FDP_ACC.2.1/WU)

(a) Subjects: *Web Users*

(b) Objects: *Content*

5.1.4.1.2. The TSF shall ensure that all operations between any subject in the **WEBUSER** TSC and any object within the **WEBUSER** TSC are covered by the **WEBUSER** SFP. (FDP_ACC.2.2/WU)

5.1.4.2. FDP_ACF.1-NIAP-0407/WU: Security Attribute Based Access Control (SFP: WEBUSER)

5.1.4.2.1. The TSF shall enforce the *WEBUSER SFP* to **controlled-access content** objects based on **the following types of subject and object security attributes:** (FDP_ACF.1.1-NIAP-0407/WU)

(a) the authorized web-user identity and group membership(s) associated with a subject and

(b) the <authorized web-user (or group) identity, access operations> pairs associated with as object.

5.1.4.2.2. The TSF shall enforce the following **WEBUSER SFP ordered** rules to determine if an operation among controlled subjects and controlled objects is allowed: (FDP_ACF.1.2-NIAP-0407/WU)

(a) For controlled-access content:

1. *If the requested access is denied to that web user, deny access.*
2. *If the requested access is something other than read access, deny access.*
3. *If read-only access is permitted to that authorized web user, grant access.*
4. *If read-only access is denied to every group of which the authorized web user is a member, deny access.*
5. *If read-only access is permitted to any group of which the authorized web user is a member, grant access.*
6. *Otherwise, deny access.*

(b) For public content

1. *If the requested access is something other than read access, deny access.*

2. *Grant read-only access to web user.*

5.1.4.2.3. The TSF shall explicitly authorize access of subjects to objects based on the following additional **WEBUSER SFP** rules: (FDP_ACF.1.3-NIAP-0407/WU)

(a) [selection: [assignment: *rules, based on security attributes, that explicitly authorize access of subjects to objects*], "*no additional rules*"]

5.1.4.2.4. The TSF shall explicitly deny access of subjects to objects based on the following additional **WEBUSER SFP** rules: (FDP_ACF.1.4-NIAP-0407/WU)

(a) [selection: [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*], "*no additional rules*"]

5.1.4.3. FDP_UCT.1/WU: Basic Data Exchange Confidentiality (SFP: WEBUSER)

5.1.4.3.1. The TSF shall enforce the *WEBUSER SFP* to be able to *transmit and receive* **controlled-access content** objects in a manner protected from unauthorized disclosure. (FDP_UCT.1.1/WU)

5.1.4.4. FDP_UIT.1/WU: Data Exchange Integrity (SFP: WEBUSER)

5.1.4.4.1. The TSF shall enforce the *WEBUSER SFP* to be able to *transmit and receive* user data in a manner protected from *modification* errors. (FDP_UIT.1.1/WU)

5.1.4.4.2. The TSF shall be able to determine on receipt of user data, **under the WEBUSER SFP**, whether *modification* has occurred. (FDP_UIT.1.1/WU)

Application Note:

The intent of the FDP_UCT and FDP_UIT elements in this SFP are to require the use of an encrypting protocol during transmission of content to which access control has been applied (i.e., controlled-access content).

5.1.5. FDP/CP: User Data Protection: Content-Provider (CP) SFP

This SFP dictates the rules that control the ability for content providers (typically, a subset of the users on the host platform) to install and modify content. Unlike typical DAC policies, this SFP is more centrally controlled, with the server administrator having control over the ability of the content providers to install and modify content.

5.1.5.1. FDP_ACC.2/CP: Complete Object Access Control (SFP: CONTENT-PROVIDER)

5.1.5.1.1. The TSF shall enforce the *CONTENT-PROVIDER SFP* on **the following subjects and objects,** and upon all operations among subjects and objects covered by **this** Security Function Policy (SFP): (FDP_ACC.2.1/CP)

(a) *Subjects: Content Providers*

(b) *Objects: Content*

5.1.5.1.2. The TSF shall ensure that all operations between any subject in the **CONTENT-PROVIDER** TSC and any object within the **CONTENT-PROVIDER** TSC are covered by the **CONTENT-PROVIDER** SFP. (FDP_ACC.2.2/CP)

5.1.5.2. FDP_ACF.1-NIAP-0407/CP: Security Attribute Based Access Control (SFP: CONTENT-PROVIDER)

5.1.5.2.1. The TSF shall enforce the *CONTENT-PROVIDER SFP* to objects based *on the identity and group membership of the content provider, the protections on the underlying objects used to create or modify content by the host platform, and the server administrative configuration.* (FDP_ACF.1.1-NIAP-0407/CP)

5.1.5.2.2. The TSF shall enforce the following **CONTENT-PROVIDER SFP** rules to determine if an operation among controlled subjects and controlled objects is allowed: (FDP_ACF.1.2-NIAP-0407/CP)

(a) *The TOE shall restrict the ability to create or modify content to only those content providers authorized by a server administrator.*

(b) *The TOE shall be capable of limiting the ability to create or modify server executable content to a subset of the authorized content providers.*

5.1.5.2.3. The TSF shall explicitly authorize access of subjects to objects based on the following additional **CONTENT-PROVIDER SFP** rules: (FDP_ACF.1.3-NIAP-0407/CP)

(a) *The TOE shall provide the ability for content providers to indicate that content is "public content", and thus accessible by any authenticated or unauthenticated web user.*

(b) [selection: [assignment: *rules, based on security attributes, that explicitly authorize access of subjects to objects*], "*no additional rules*"]

5.1.5.2.4. The TSF shall explicitly deny access of subjects to objects based on the following additional **CONTENT-PROVIDER SFP** rules: (FDP_ACF.1-NIAP-0407.4/CP)

(a) [selection: [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*], "*no additional rules*"]

5.1.6. FDP: Other User Data Protection Policies

5.1.6.1. FDP_RIP.2: Full Residual Information Protection

5.1.6.1.1. The TSF shall ensure that any previous information content of a resource is made unavailable upon the deallocation of **all objects**. (FDP_RIP.2.1)

5.1.7. FIA: Identification and authentication

5.1.7.1. FIA_AFL.1-NIAP-0425: Authentication failure handling

5.1.7.1.1. The TSF shall detect when [*an administrator configurable integer*] **of** unsuccessful authentication attempts occur related to *web user login and content provider login (if the content provider login is provided by the TSF)***.** (FIA_AFL.1.1-NIAP-0425)

5.1.7.1.2. When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall *prevent the [assignment: entities requesting authentication] from performing activities that require authentication until an action is taken by the administrator*. (FIA_AFL.1.2-NIAP-0425)

Application Note:

The Web Server Administrator is exempted from this requirement to avoid an administrative denial of service.

Unsuccessful authentication refers to the presenting of invalid authentication information. This includes invalid passwords as well as invalid certificates.

5.1.7.2. FIA_ATD.1: User Attribute Definition

5.1.7.2.1. The TSF shall maintain the following list of security attributes belonging to **each web user and content provider**: (FIA_ATD.1.1)

(a) *Identification of the user*

(b) *Credentials used to authenticate the user*

Operations Note:

This was refined to clarify the meaning of the phrase “individual users” to be those users within the TSC.

5.1.7.3. FIA_UAU.1: Timing of Authentication

5.1.7.3.1. The TSF shall allow *read-only access of content designated as public* on behalf of **a web user** to be performed before the **web** user is authenticated. (FIA_UAU.1.1)

5.1.7.3.2. The TSF shall require each **web user, content provider, and administrator** to **have been** successfully authenticated before allowing any TSF-mediated actions on behalf of that user. (FIA_UAU.1.2)

Application Note:

If the underlying IT environment provides authentication services for content providers and administrators, it is acceptable for FIA_UAU.1.2 to be satisfied by the presentation and verification of those credentials.

5.1.7.4. FIA_UAU.7: Protected Authentication Feedback

5.1.7.4.1. The TSF shall provide feedback that *does not disclose passwords or private keys* to the user while the authentication is in progress. (FIA_UAU.7.1)

5.1.7.5. FIA_UID.1: Timing of Identification

5.1.7.5.1. The TSF shall allow *only access of content designated as public* on behalf of **a web user** to be performed before the **web** user is identified. (FIA_UID.1.1)

5.1.7.5.2. The TSF shall require each **web user, content provider, and administrator** to **have been** successfully identified before allowing any TSF-mediated actions on behalf of that user. (FIA_UID.1.2)

Application Note:

If the underlying IT environment provides identification services for content providers and administrators, it is acceptable for FIA_UID.1.2 to be satisfied by the presentation and verification of those credentials.

5.1.7.6. FIA_USB.1-NIAP-0351: User-Subject Binding

5.1.7.6.1. The TSF shall associate all user security attributes with subjects acting on behalf of that user. (FIA_USB.1.1-NIAP-0351)

5.1.8. FMT: Security management

5.1.8.1. FMT_MOF.1: Management of Security Functions Behavior

5.1.8.1.1. The TSF shall restrict the ability to **perform the following functions** to the *authorized TOE administrator*: (FMT_MOF.1.1)

(a) enable, disable, and modify the TOE audit functions

(b) *delete stored audit records*

(c) *include or exclude auditable events from the set of audited events*

(d) *adjust the web server configuration parameters*

Operation Notes:

This was refined to make the list of restricted functions a list.

5.1.8.2. FMT_MSA.1 Management of security attributes

5.1.8.2.1. The TSF shall enforce the *WEBUSER and CONTENT-PROVIDER SFPs* to restrict the ability to [selection: change_default, query, modify, delete, [assignment: other operations]] the security attributes [assignment: list of security attributes] to [assignment: the authorized identified roles]. (FMT_MSA.1.1)

5.1.8.3. FMT_MSA.2: Secure Security Attributes

5.1.8.3.1. The TSF shall ensure that only secure values are accepted for security attributes. (FMT_MSA.2.1)

5.1.8.4. FMT_MSA.3-NIAP-0429: Static attribute initialization

5.1.8.4.1. The TSF shall enforce the *WEBUSER and CONTENT-PROVIDER SFPs* to provide *restrictive* default values for security attributes that are used to enforce the SFP. (FMT_MSA.3.1-NIAP-0429)

5.1.8.4.2. The TSF shall allow the *Web Server Administrator* to specify alternative initial values to override the default values when an object or information is created. (FMT_MSA.3.2-NIAP-0429)

5.1.8.5. FMT_MTD.1: Management of TSF Data

5.1.8.5.1. The TSF shall restrict the ability *to change the default, query, modify, delete, clear, and define* the *TOE content* to the *Web Server Administrator and Content Providers*. (FMT_MTD.1.1)

5.1.8.6. FMT_REV.1: Revocation

5.1.8.6.1. The TSF shall restrict the ability to revoke security attributes associated with the *web users, content providers, and controlled objects* within the TSC to *Web Server Administrator*. (FMT_REV.1.1)

5.1.8.6.2. The TSF shall enforce the **following** rules**:** [assignment: *specification of revocation rules*]. (FMT_REV.1.2)

5.1.8.7. FMT_SMR.1: Security Roles

5.1.8.7.1. The TSF shall maintain the following roles: (FMT_SMR.1.1)

(a) *Web Server Administrator*

(b) *Content Provider*

(c) *Web User*

5.1.8.7.2. The TSF shall be able to associate users with roles. (FMT_SMR.1.2)

5.1.9. FPT: Protection of the TOE Security Functions

5.1.9.1. FPT_AMT.1 Abstract machine testing

5.1.9.1.1. The TSF shall run a suite of tests [selection: during initial start-up, periodically during normal operation, at the request of an authorized user, other conditions] to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF. (FPT_AMT.1.1)

5.1.9.2. FPT_RCV.2 Automated Recovery

5.1.9.2.1. When automated recovery from a failure or service discontinuity is not possible, the TSF shall enter a maintenance mode where the ability to return the TOE to a secure state is provided. (FPT_RCV.2.1)

5.1.9.2.2. For *power disruptions, interface failures, or software failures that result in the TOE faulting to an unsecured state*, the TSF shall ensure the return of the TOE to a secure state using automated procedures. (FPT_RCV.2.2)

Application Note:

The dependency on ADV_SPM.1 is not satisfied, because discussion within the CCIMB has indicated that this is an erroneous dependency, and will be removed in an upcoming interpretation.

5.1.9.3. FPT_RVM.1: Non-bypassability of the TSP

5.1.9.3.1. The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed. (FPT_RVM.1.1)

5.1.9.4. FPT_SEP_EXP.1: Application Domain Separation

5.1.9.4.1. The TSF shall maintain a security domain that protects it from interference and tampering by untrusted subjects initiating actions through its own TSFI. (FPT_SEP_EXP.1.1)

5.1.9.4.2. The TSF shall enforce separation between the security domains of subjects in the TOE Scope of Control. (FPT_SEP_EXP.1.2)

5.1.9.5. FPT_STM.1 Reliable time stamps

5.1.9.5.1. The TSF shall be able to provide reliable time stamps for its own use. (FPT_STM.1.1)

5.1.9.6. FPT_TST.1/CR: TSF Testing (Cryptography and Critical Functions)

5.1.9.6.1. The TSF shall run a suite of self-tests**, at the following times, in accordance with FIPS PUB 140-2, Level 1 (as identified in Table 5-1)** to demonstrate the correct operation of the **indicated functions of the** TOE. (FPT_TST.1.1/CR)

(a) *Testing Times: during initial start-up (on power on); at the request of the administrator (on demand); under the following conditions* [assignment: *other conditions under which the cryptographic self tests shall be run*]*; and periodically.*

(b) *Functions to be tested: cryptographically software/firmware; cryptographic algorithms; RNG/PRNG; other FIPS PUB 140-2 critical functions; and* [assignment: *list of all critical security functions*].

Application Note:

The ST author fills in the conditions under which the self-tests are run by consulting FIPS 140-2 as well as to reflect capabilities of the TOE.

Table 5-4. Interpretation of FIPS PUB 140-2 Self Tests

Self-Tests	FIPS-140 Security Level 1
Software/Firmware Integrity Tests	on power on on demand conditional
Cryptographic Algorithm Tests	on power on on demand conditional

Self-Tests	FIPS-140 Security Level 1
Other FIPS PUB 140-2 critical functions tests and other tests as determined by FIPS PUB 140-2, Appendix A	on power on on demand conditional
Statistical RNG/PRNG tests	on power on on demand

-

5.1.9.6.2. The TSF shall provide **the administrators** with the capability to verify the integrity of **cryptographically related** TSF data. (FPT_TST.1.2/CR)

5.1.9.6.3. The TSF shall provide **the administrators** with the capability to verify the integrity of stored **cryptographically related** TSF executable code. (FPT_TST.1.3/CR)

5.1.9.7. FPT_TST.1/NC: TSF Testing (Non-Cryptographic Code)

5.1.9.7.1. The TSF shall run a suite of set tests *at the request of the authorized user* to demonstrate the correct operation of the **non-cryptographic portions of** the TOE. (FPT_TST.1.1/NC)

5.1.9.7.2. The TSF shall provide authorized users with the capability to verify the integrity of the **non-cryptographic** TSF data. (FPT_TST.1.2/NC)

5.1.9.7.3. The TSF shall provide authorized users with the capability to verify the integrity of stored **non-cryptographic** TSF code. (FPT_TST.1.3/NC)

5.1.9.8. FPT_TST_EXP.1/KG: TSF Testing (Key Generation Components)

5.1.9.8.1. The TSF shall run a suite of self-tests immediately after generation of a key to demonstrate correct operation of each key generation component. If any of these tests fails, that generated key shall not be used, the cryptographic module shall react as required by FIPS PUB 140 for failing a self-test, and this event will be audited. (FPT_TST_EXP.1.1/KG)

Application Note:

Key generation components are those critical elements that compose the entire key generation process (e.g., any algorithms, any RNG/PRNGs, any key generation seeding processes, etc.).

5.1.9.8.2. The TSF shall provide the Administrator with the capability to verify the integrity of TSF data related to the key generation. (FPT_TST_EXP.1.2/KG)

5.1.9.8.3. The TSF shall provide the Administrator with the capability to verify the integrity of stored TSF executable code related to the key generation. (FPT_TST_EXP.1.3/KG)

5.1.10. FTA: TOE Access

5.1.10.1. FTA_SSL.1: TSF-initiated session locking

5.1.10.1.1. The TSF shall lock a local interactive session after *a Web Server Administrator-specified time period of inactivity* by: (FTA_SSL.1.1)

(a) clearing or overwriting display devices, making the current contents unreadable.

(b) disabling any activity of the user's data access/display devices other than unlocking the session.

5.1.10.1.2. The TSF shall require the following events to occur prior to unlocking the session: *reauthentication by the administrative user***.** (FTA_SSL.1.2)

5.1.10.2. FTA_SSL.2: User-initiated locking

5.1.10.2.1. The TSF shall allow user-initiated locking of the **administrative** user's own local interactive session by: (FTA_SSL.2.1)

(c) clearing or overwriting display devices, making the current contents unreadable.

(d) disabling any activity of the user's data access/display devices other than unlocking the session.

5.1.10.2.2. The TSF shall require the following events to occur prior to unlocking the session: *reauthentication by the administrative user***.** (FTA_SSL.2.2)

Application Note:

The interactive sessions in FTA_SSL.1 and FTA_SSL.2 are those of the local web server administrator. Non-administrators only have remote access to the TOE and the requirements for session locking levied on them are specified in FTA_SSL.3.

5.1.10.3. FTA_SSL.3/IN: TSF-initiated termination

5.1.10.3.1. The TSF shall terminate a **remote interactive** session after a [*web server Administrator-configurable time interval of session inactivity*]. (FTA_SSL.3.1/IN)

Application Note:

A remote interactive session applies to remote web server administrators.

5.1.10.4. FTA_SLL.3/WU: Web User Termination

5.1.10.4.1. The TSF shall **request revalidation of user credentials** after a [*web server Administrator-configurable time internal of session inactivity*]. (FTA_SSL.3.1/WU)

Application Note:

HTTP and HTTP are datagram oriented, not session oriented protocols. Thus, technically, one cannot terminate a session. However, one can have the same effect by requiring the user to revalidate their credentials, as opposed to using the cached or forwarded credentials.

5.1.10.5. FTA_TAB.1: Default TOE Access Banners

Before establishing an **administrative** user session, the TSF shall display an advisory warning message regarding unauthorized use of the TOE. (FTA_TAB.1.1)

Application Note:

This has been restricted to administrative user sessions. Web user (and content provider, through HTTP) access has screens that are not under control of the web server, but under control of the content provider, and thus, outside the TSC.

5.1.11. FTP: Trusted Path

5.1.11.1. FTP_ITC.1 Inter-TSF trusted channel

5.1.11.1.1. The TSF shall provide a communication channel between itself and a remote trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure. (FTP_ITC.1.1)

5.1.11.1.2. The TSF shall permit *either the TSF or the remote trusted IT product* to initiate communication via the trusted channel. (FTP_ITC.1.2)

5.1.11.1.3. The TSF shall initiate communication via the trusted channel for *the transmission of controlled-access content*. (FTP_ITC.1.3)

5.2. IT Environment Security Functional Requirements

The notional model is that the Web Server is a software application built on top of an underlying IT platform. This IT platform provides basic controlled access services such as identification and authentication, discretionary access control, residual information protection, protection for the TOE, and a basic level of robustness. Instead of duplicating an already existing profile in this document, the approach taken is to require that the underlying platform be compliant with an appropriate profile. Note that it is acceptable for the TOE to satisfy IT environment requirements; this would be captured in the ST.

5.2.1. FIT_PPC_EXP: IT Environment Profile Compliance

5.2.1.1. FIT_PPC_EXP: IT Environment Profile Compliance

5.2.1.1.1. The IT Environment shall be compliant with the requirements of the Controlled Access Protection Profile or an Operating System Protection Profile at the Basic Level of Robustness or Greater. (FIT_PPC_EXP.1.1)

Application Note:

This requirement can be met by providing evidence (e.g., certificate) that the underlying operating system is compliant with the Controlled Access Protection Profile or with a protection profile at the Basic Level of Robustness or greater.

5.3. TOE Security Assurance Requirements

The assurance requirements levied on the developer consist of EAL2 Augmented (augmentations are shown in bold) and are summarized in Table 5-5.

Table 5-5. EAL 2 Assurance Requirements

Assurance Class	Assurance Components	
Configuration management	ACM_CAP.2	Configuration items
Delivery and operation	ADO_DEL.1	Delivery procedures
	ADO_IGS.1	Installation, generation, and start-up procedures
Development	ADV_FSP.1	Informal functional specification
	ADV_HLD.1	Descriptive high-level
	ADV_RCR.1	Informal Correspondence Demonstration
Guidance documents	AGD_ADM.1	Administrator guidance
	AGD_USR.1	User guidance
Tests	ATE_COV.1	Test Coverage
	ATE_FUN.1	Functional testing
	ATE_IND.2	Independent testing – sample

Assurance Class	Assurance Components	
Life Cycle	**ALC_FLR.2**	**Flaw Remediation**
Vulnerability assessment	**AVA_MSU.1**	**Misuse – Examination of Guidance**
	AVA_SOF.1	Strength of TOE security function evaluation
	AVA_VLA.1	Developer vulnerability analysis
Maintenance of Assurance	**AMA_AMP.1**	**Assurance maintenance plan**
	AMA_CAT.1	**TOE component categorization report**
	AMA_EVD.1	**Evidence of assurance maintenance**
	AMA_SIA.1	**Security impact analysis**

5.3.1. ACM: Configuration Management

5.3.1.1. ACM_CAP.2: Configuration Items

Developer action elements:

5.3.1.1.1. The developer shall provide a reference for the TOE. (ACM_CAP.2.1D)

5.3.1.1.2. The developer shall provide CM documentation. (ACM_CAP.2.3D)

Content and presentation of evidence elements:

5.3.1.1.3. The reference for the TOE shall be unique to each version of the TOE. (ACM_CAP.2.1.C)

5.3.1.1.4. The TOE shall be labeled with its reference. (ACM_CAP.2.2C)

5.3.1.1.5. The CM documentation shall include a configuration list. (ACM_CAP.2.3C)

5.3.1.1.6. The configuration list shall describe the configuration items that comprise the TOE. (ACM_CAP.2.4C)

5.3.1.1.7. The CM documentation shall describe the method used to uniquely identify the configuration items. (ACM_CAP.2.5C)

5.3.1.1.8. The CM system shall uniquely identify all configuration items. (ACM_CAP.2.6C-NIAP-0412)

Evaluator action elements:

5.3.1.1.9. The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence. (ACM_CAP.2.1E)

5.3.2. ADO: Delivery and Operation

5.3.2.1. ADO_DEL.1: Delivery Procedures

Developer action elements:

5.3.2.1.1. The developer shall document procedures for delivery of the TOE or parts of it to the user. (ADO_DEL.1.1D)

5.3.2.1.2. The developer shall use the delivery procedures. (ADO_DEL.1.2D)

Content and presentation of evidence elements:

5.3.2.1.3. The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site. (ADO_DEL.1.1C)

Evaluator action elements:

5.3.2.1.4. The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence. (ADO_DEL.1.1E)

5.3.2.2. ADO_IGS.1: Installation, Generation, and Start-Up Procedures (ADO_IGS.1)

Developer action elements:

5.3.2.2.1. The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE. (ADO_IGS.1.1D)

Content and presentation of evidence elements:

5.3.2.2.2. The documentation shall describe the steps necessary for secure installation, generation, and start-up of the TOE. (ADO_IGS.1.1C)

Evaluator action elements:

5.3.2.2.3. The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence (ADO_IGS.1.1E)

5.3.2.2.4. The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration. (ADO_IGS.1.2E)

5.3.3. ADV: Development

5.3.3.1. ADV_FSP.1: Informal Functional Specification

Developer action elements:

5.3.3.1.1. The developer shall provide a functional specification. [(ADV_FSP.1.1D)]

Content and presentation of evidence elements:

5.3.3.1.2. The functional specification shall describe the TSF and its external interfaces using an informal style. [(ADV_FSP.1.1C)]

5.3.3.1.3. The functional specification shall be internally consistent. [(ADV_FSP.1.2C)]

5.3.3.1.4. The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate. [(ADV_FSP.1.3C)]

5.3.3.1.5. The functional specification shall completely represent the TSF. [(ADV_FSP.1.4C)]

Evaluator action elements:

5.3.3.1.6. The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence. [(ADV_FSP.1.1E)]

5.3.3.1.7. The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements. [(ADV_FSP.1.2E)]

5.3.3.2. ADV_HLD.1: Descriptive High-Level Design

Developer action elements:

5.3.3.2.1. The developer shall provide the high-level design of the TSF. [(ADV_HLD.1.1D)]

Content and presentation of evidence elements:

5.3.3.2.2. The presentation of the high-level design shall be informal. (ADV_HLD.1.1C)

5.3.3.2.3. The high-level design shall be internally consistent. (ADV_HLD.1.2C)

5.3.3.2.4. The high-level design shall describe the structure of the TSF in terms of subsystems. (ADV_HLD.1.3C)

5.3.3.2.5. The high-level design shall describe the security functionality provided by each subsystem of the TSF. (ADV_HLD.1.4C)

5.3.3.2.6. The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software. (ADV_HLD.1.5C)

5.3.3.2.7. The high-level design shall identify all interfaces to the subsystems of the TSF. (ADV_HLD.1.6C)

5.3.3.2.8. The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible. (ADV_HLD.1.7C)

Evaluation action elements

5.3.3.2.9. The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence (ADV_HLD.1.1E)

5.3.3.2.10. The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements. (ADV_HLD.1.2E)

5.3.3.3. ADV_RCR.1: Informal Correspondence Demonstration

Developer action elements:

5.3.3.3.1. The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided. (ADV_RCR.1.1D)

Content and presentation of evidence elements:

5.3.3.3.2. For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation. (ADV_RCR.1.1C)

Evaluator action element:

5.3.3.3.3. The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence. (ADV_RCR.1.1E)

5.3.4. AGD: Guidance Documents

5.3.4.1. AGD_ADM.1: Administrator Guidance

Developer action elements:

5.3.4.1.1. The developer shall provide administrator guidance addressed to system administrative personnel. (AGD_ADM.1.1D)

Content and presentation of evidence elements:

5.3.4.1.2. The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE. (AGD_ADM.1.1C)

5.3.4.1.3. The administrator guidance shall describe how to administer the TOE in a secure manner. (AGD_ADM.1.2C)

5.3.4.1.4. The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment. (AGD_ADM.1.3C)

5.3.4.1.5. The administrator guidance shall describe all assumptions regarding user behavior that are relevant to secure operation of the TOE. (AGD_ADM.1.4C)

5.3.4.1.6. The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate. (AGD_ADM.1.5C)

5.3.4.1.7. The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF. (AGD_ADM.1.6C)

5.3.4.1.8. The administrator guidance shall be consistent with all other documentation supplied for evaluation. (AGD_ADM.1.7C)

5.3.4.1.9. The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator. (AGD_ADM.1.8C)

Evaluator action elements:

5.3.4.1.10. The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence. (AGD_ADM.1.1E)

5.3.4.2. AGD_USR.1: User Guidance

Developer action elements:

5.3.4.2.1. The developer shall provide user guidance. (AGD_USR.1.1D)

Content and presentation of evidence elements:

5.3.4.2.2. The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE. (AGD_USR.1.1C)

5.3.4.2.3. The user guidance shall describe the use of user-accessible security functions provided by the TOE. (AGD_USR.1.2C)

5.3.4.2.4. The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment. (AGD_USR.1.3C)

5.3.4.2.5. The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behavior found in the statement of TOE security environment. (AGD_USR.1.4C)

5.3.4.2.6. The user guidance shall be consistent with all other documentation supplied for evaluation. (AGD_USR.1.5C)

5.3.4.2.7. The user guidance shall describe all security requirements for the IT environment that are relevant to the user. (AGD_USR.1.6C)

Evaluator action elements:

5.3.4.2.8. The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence. (AGD_USR.1.1E)

5.3.5. ALC: Life Cycle Support

5.3.5.1. ALC_FLR.1: Flaw Remediation

Developer action elements:

5.3.5.1.1. The developer shall document the flaw remediation procedures. (ALC_FLR.1.1D)

Content and presentation of evidence elements:

5.3.5.1.2. The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE. (ALC_FLR.1.1C)

5.3.5.1.3. The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw. (ALC_FLR.1.2C)

5.3.5.1.4. The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws. (ALC_FLR.1.3C)

5.3.5.1.5. The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users. (ALC_FLR.1.4C)

Evaluator action elements:

5.3.5.1.6. The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence. (ALC_FLR.1.1E)

5.3.6. ATE: Tests

5.3.6.1. ATE_COV.1: Evidence of Coverage

Developer action elements:

5.3.6.1.1. The developer shall provide evidence of the test coverage. (ATE_COV.1.1D)

Content and presentation of evidence elements:

5.3.6.1.2. The evidence of the test coverage shall show the correspondence between tests identified in the test documentation and the TSF as described in the functional specification. (ATE_COV.1.1C)

Evaluator action elements:

5.3.6.1.3. The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence. (ATE_COV.1.1E)

5.3.6.2. ATE_FUN.1: Functional Testing

Developer action elements:

5.3.6.2.1. The developer shall test the TSF and document the results. (ATE_FUN.1.1D)

5.3.6.2.2. The developer shall provide test documentation. (ATE_FUN.1.2D)

Content and presentation of evidence elements:

5.3.6.2.3. The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results. (ATE_FUN.1.1C)

5.3.6.2.4. The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed. (ATE_FUN.1.2C)

5.3.6.2.5. The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests. (ATE_FUN.1.3C)

5.3.6.2.6. The expected test results shall show the anticipated outputs from a successful execution of the tests. (ATE_FUN.1.4C)

5.3.6.2.7. The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified. (ATE_FUN.1.5C)

Evaluator action elements:

5.3.6.2.8. The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence. (ATE_FUN.1.1E)

5.3.6.3. ATE_IND.2: Independent Testing - Sample

Developer action elements:

5.3.6.3.1. The developer shall provide the TOE for testing. (ATE_IND.2.1D)

Content and presentation of evidence elements:

5.3.6.3.2. The TOE shall be suitable for testing. (ATE_IND.2.1C)

5.3.6.3.3. The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF. (ATE_IND.2.2C)

Evaluator action elements:

5.3.6.3.4. The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence. (ATE_IND.2.1E)

5.3.6.3.5. The evaluator shall test a subset of the TSF as appropriate to confirm that the TOE operates as specified. (ATE_IND.2.2E)

5.3.6.3.6. The evaluator shall execute a sample of tests in the test documentation to verify the developer test results. (ATE_IND.2.3E)

5.3.7. AVA: Vulnerability Assessment

5.3.7.1. AVA_MSU.1: Examination of guidance

Developer action elements:

5.3.7.1.1. The developer shall provide guidance documentation. (AVA_MSU.1.1D)

Content and presentation of evidence elements:

5.3.7.1.2. The guidance documentation shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation. (AVA_MSU.1.1C)

5.3.7.1.3. The guidance documentation shall be complete, clear, consistent and reasonable. (AVA_MSU.1.2C)

5.3.7.1.4. The guidance documentation shall list all assumptions about the intended environment. (AVA_MSU.1.3C)

5.3.7.1.5. The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls). (AVA_MSU.1.4C)

Evaluator action elements:

5.3.7.1.6. The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence. (AVA_MSU.1.1E)

5.3.7.1.7. The evaluator shall repeat all configuration and installation procedures to confirm that the TOE can be configured and used securely using only the supplied guidance documentation. (AVA_MSU.1.2E)

5.3.7.1.8. The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected. (AVA_MSU.1.3E)

5.3.7.2. AVA_SOF.1: Strength of TOE Security Function Evaluation

Developer action elements:

5.3.7.2.1. The developer shall perform a strength of TOE security function analysis for each mechanism identified in the ST as having a strength of TOE security function claim. (AVA_SOF.1.1D)

Content and presentation of evidence elements:

5.3.7.2.2. For each mechanism with a Strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST. (AVA_SOF.1.1C)

Application Note:

See Table 5-6 for the recommended strength of function claims.

Table 5-6. Recommended Strength of Function Claims

Mechanism	Minimum Strength Level	Strength of Function Metric
Password used for access control under the WEBUSER SFP	SOF-Medium	None
Certificates	SOF-Medium	None

5.3.7.2.3. For each mechanism with a specific strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the PP/ST. (AVA_SOF.1.2C)

Evaluator action elements:

5.3.7.2.4. The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence. (AVA_SOF.1.1E)

5.3.7.2.5. The evaluator shall confirm that the strength claims are correct. (AVA_SOF.1.2E)

5.3.7.3. AVA_VLA.1: Developer Vulnerability Analysis

Developer action elements:

5.3.7.3.1. The developer shall perform and document an analysis of the TOE deliverables searching for obvious ways in which a user can violate the TSP. (AVA_VLA.1.1D)

5.3.7.3.2. The developer shall document the disposition of obvious vulnerabilities. (AVA_VLA.1.2D)

Content and presentation of evidence elements:

5.3.7.3.3. The documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE. (AVA_VLA.1.1C)

Evaluator action elements:

5.3.7.3.4. The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence. (AVA_VLA.1.1E)

5.3.7.3.5. The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure obvious vulnerabilities have been addressed. (AVA_VLA.1.2E)

5.3.8. AMA: Maintenance of Assurance

5.3.8.1. AMA_AMP.1 Assurance maintenance plan

Developer action elements:

5.3.8.1.1. The developer shall provide an AM Plan. (AMA_AMP.1.1D)

Content and presentation of evidence elements:

5.3.8.1.2. The AM Plan shall contain or reference a brief description of the TOE, including the security functionality it provides.(AMA_AMP.1.2C)

5.3.8.1.3. The AM Plan shall reference the TOE component categorization report for the certified version of the TOE.(AMA_AMP.1.3C)

5.3.8.1.4. The AM Plan shall describe the scope of changes to the TOE that are covered by the plan.(AMA_AMP.1.4C)

5.3.8.1.5. The AM Plan shall describe the TOE life-cycle, and shall identify the current plans for any new releases of the TOE, together with a brief description of any planned changes that are likely to have a significant security impact.(AMA_AMP.1.5C)

5.3.8.1.6. The AM Plan shall describe the assurance maintenance cycle, stating and justifying the planned schedule of AM audits and the target date of the next re-evaluation of the TOE. (AMA_AMP.1.6C)

5.3.8.1.7. The AM Plan shall identify the individual(s) who will assume the role developer security analyst of the TOE. (AMA_AMP.1.7C)

5.3.8.1.8. The AM Plan shall describe how the developer security analyst role will ensure that the procedures documented or referenced in the AM Plan are followed.(AMA_AMP.1.8C)

5.3.8.1.9. The AM Plan shall describe how the developer security analyst role will ensure that all developer actions involved in the analysis of the security impact of changes affecting the TOE are performed correctly.(AMA_AMP.1.9C)

5.3.8.1.10. The AM Plan shall justify why the identified developer security analyst(s) have sufficient familiarity with the security target, functional specification and (where appropriate) high-level design of the TOE, and with the evaluation results and all applicable assurance requirements for the certified version of the TOE.(AMA_AMP.1.10C)

5.3.8.1.11. The AM Plan shall describe or reference the procedures to be applied to maintain the assurance in the TOE, which as a minimum shall include the procedures for configuration management, maintenance of assurance evidence, performance of the analysis of the security impact of changes affecting the TOE, and flaw remediation.(AMA_AMP.1.11 C)

Evaluator action elements:

5.3.8.1.12. The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.(AMA_AMP.1.1E)

5.3.8.1.13. The evaluator shall confirm that the proposed schedules for AM audits and re-evaluation of the TOE are acceptable and consistent with the proposed changes to the TOE. (AMA_AMP.12E)

5.3.8.2. AMA_CAT.1 TOE component categorization report

Developer action elements:

5.3.8.2.1. The developer shall provide a TOE component categorization report for the certified version of the TOE.(AMA_CAT1.1D)

Content and presentation of evidence elements:

5.3.8.2.2. The TOE component categorization report shall categorize each component of the TOE, identifiable in each TSF representation from the most abstract to the least abstract, according to its relevance to security; as a minimum, TOE components must be categorized as one of TSP-enforcing or non-TSP- enforcing.(AMA_CAT1.1C)

5.3.8.2.3. The TOE component categorization report shall describe the categorization scheme used, so that it can be determined how to categorize new components introduced into the TOE, and also when to re-categorize existing TOE components following changes to the TOE or its security target.(AMA_CAT1.2C)

5.3.8.2.4. The TOE component categorization report shall identify any tools used in the development environment that, if modified, will have an impact on the assurance that the TOE satisfies its security target.(AMA_CAT1.2C)

Evaluator action elements:

5.3.8.2.5. The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.(AMA_CAT1.1E)

5.3.8.2.6. The evaluator shall confirm that the categorization of TOE components and tools, and the categorization scheme used, are appropriate and consistent with the evaluation results for the certified version.(AMA_CAT1.2E)

5.3.8.3. AMA_EVD.1 Evidence of maintenance process

Developer action elements:

5.3.8.3.1. The developer security analyst shall provide AM documentation for the current version of the TOE.(AMA_EVD1.1D)

Content and presentation of evidence elements:

5.3.8.3.2. The AM documentation shall include a configuration list and a list of identified vulnerabilities in the TOE.(AMA_EVD1.1C)

5.3.8.3.3. The configuration list shall describe the configuration items that comprise the current version of the TOE.(AMA_EVD1.2C)

5.3.8.3.4. The AM documentation shall provide evidence that the procedures documented or referenced in the AM Plan are being followed.(AMA_EVD1.3C)

5.3.8.3.5. The list of identified vulnerabilities in the current version of the TOE shall show, for each vulnerability, that the vulnerability cannot be exploited in the intended environment for the TOE.(AMA_EVD1.4C)

Evaluator action elements:

5.3.8.3.6. The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.(AMA_EVD1.1E)

5.3.8.3.7. The evaluator shall confirm that the procedures documented or referenced in the AM Plan are being followed.(AMA_EVD1.2E)

5.3.8.3.8. The evaluator shall confirm that the security impact analysis for the current version of the TOE is consistent with the configuration list.(AMA_EVD1.3E)

5.3.8.3.9. The evaluator shall confirm that all changes documented in the security impact analysis for the current version of the TOE are within the scope of changes covered by the AM Plan.(AMA_EVD1.4E)

5.3.8.3.10. The evaluator shall confirm that functional testing has been performed on the current version of the TOE, to a degree commensurate with the level of assurance being maintained. (AMA_EVD1.5E)

5.3.8.4. AMA_SIA.1 Sampling of security impact analysis

Developer action elements:

5.3.8.4.1. The developer security analyst shall, for the current version of the TOE, provide a security impact analysis that covers all changes affecting the TOE as compared with the certified version. (AMA_SIA.1.1D)

Content and presentation of evidence elements:

5.3.8.4.2. The security impact analysis shall identify the certified TOE from which the current version of the TOE was derived. (AMA_SIA.1.1C)

5.3.8.4.3. The security impact analysis shall identify all new and modified TOE components that are categorized as TSP-enforcing. (AMA_SIA.1.2C)

5.3.8.4.4. The security impact analysis shall, for each change affecting the security target or TSF representations, briefly describe the change and any effects it has on lower representation levels. (AMA_SIA.1.3C)

5.3.8.4.5. The security impact analysis shall, for each change affecting the security target or TSF representations, identify all IT security functions and all TOE components categorized as TSP-enforcing that are affected by the change. (AMA_SIA.1.4C)

5.3.8.4.6. The security impact analysis shall, for each change which results in a modification of the implementation representation of the TSF or the IT environment, identify the test evidence that shows, to the required level of assurance, that the TSF continues to be correctly implemented following the change. (AMA_SIA.1.5C)

5.3.8.4.7. The security impact analysis shall, for each applicable assurance requirement in the configuration management (Class ACM Configuration management), life cycle support (Class ALC Life cycle support), delivery and operation (Class ADO Delivery and operation) and guidance documents (Class AGD Guidance documents) assurance classes, identify any evaluation deliverables that have changed, and provide a brief description of each change and its impact on assurance. (AMA_SIA.1.6C)

5.3.8.4.8. The security impact analysis shall, for each applicable assurance requirement in the vulnerability assessment (Class AVA Vulnerability assessment) assurance class, identify which evaluation deliverables have changed and which have not, and give reasons for the decision taken as to whether or not to update the deliverable. (AMA_SIA.1.7C)

Evaluator action elements:

5.3.8.4.9. The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence. (AMA_SIA.1.1E)

5.3.8.4.10. The evaluator shall check, by sampling, that the security impact analysis documents changes to an appropriate level of detail, together with appropriate justifications that assurance has been maintained in the current version of the TOE. (AMA_SIA.1.2E)

6.0. RATIONALE

This section describes the rationale for the Security Objectives and Security Functional Requirements as defined in Section 4.0 and Section 5.0, respectively. Additionally, this section describes the rationale for not satisfying all of the dependencies and the rationale for the strength of function (SOF) claim. Table 6-1 illustrates the mapping from Security Objectives to Threats and Policies.

6.1. Rationale for TOE Security Objectives

Table 6-1. Mapping from Threats and Policies to Security Objectives

Threat/Policy	Objectives Addressing the Threat and Policies	Rationale
T.ACCIDENTAL_ADMIN_ ERROR: An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.	**O.ADMIN_GUIDANCE**: The TOE will provide administrators with the necessary information for secure management.	**O.ADMIN_GUIDANCE** helps to mitigate this threat by ensuring the TOE administrators have guidance that instructs them how to administer the TOE in a secure manner. Having this guidance helps to reduce the mistakes that an administrator might make that could cause the TOE to be configured in a way that is unsecure.
T.ACCIDENTAL_AUDIT_ COMPROMISE: A user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's action.	**O.AUDIT_PROTECTION**: The TOE will provide the capability to protect audit information. **O.RESIDUAL_ INFORMATION**: The TOE will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated. **O.PARTIAL_SELF_PROTECTION**: The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure through its own interfaces.	**O.AUDIT_PROTECT** contributes to mitigating this threat by controlling access to the audit trail. Only the System Administrator is allowed to read the audit trail, no one is allowed to modify audit records, the System Administrator is the only one allowed to delete the audit trail, and the TOE has the capability to prevent auditable actions from occurring if the audit trail is full. **O.RESIDUAL_INFORMATION** prevents a user not authorized to read the audit trail from access to audit information that might otherwise be persistent in a TOE resource (e.g., memory). By ensuring the TOE prevents residual information in a resource, audit information will not become available to any user or process except those explicitly authorized for that data. **O.PARTIAL_SELF_PROTECTION** contributes to countering this threat by ensuring that the TSF can protect itself from users. If the TSF could not maintain and control its domain of execution, it could not be trusted to control access to the resources under its control, which includes the audit trail.

Threat/Policy	Objectives Addressing the Threat and Policies	Rationale
T.MASQUERADE: A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources.	**O.TOE_ACCESS**: The TOE will provide mechanisms that control a user's logical access to the TOE. **O.USER_CONFIDENCE** The TOE will provide mechanisms that permit web users to have confidence that received controlled-access data comes from the TOE.	**O.TOE_ACCESS** mitigates this threat by controlling the logical access to the TOE and its resources. By constraining how and when authorized users can access the TOE, and by mandating the type and strength of the authentication mechanism this objective helps mitigate the possibility of a user attempting to login and masquerade as an authorized user. In addition, this objective provides the administrator the means to control the number of failed login attempts a user can generate before an account is locked out, further reducing the possibility of a user gaining unauthorized access to the TOE. **O.USER_CONFIDENCE** also mitigates this threat by providing web users with a mechanism that assures them the data the receive isn't coming from a masqueraded entity.

Threat/Policy	Objectives Addressing the Threat and Policies	Rationale
T.POOR_DESIGN: Unintentional errors in requirements specification or design of the TOE may occur, leading to flaws that may be exploited by a casually mischievous user or program.	**O.CONFIGURATION_IDENTIFICATION**: The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified, corrected with the TOE being redistributed promptly. **O.DOCUMENTED_DESIGN**: The design of the TOE is adequately and accurately documented. **O.VULNERABILITY_ANALYSIS**: The TOE will undergo some vulnerability analysis demonstrate the design and implementation of the TOE does not contain any obvious flaws.	**O.CONFIGURATION_IDENTIFI-CATION** plays a role in countering this threat by requiring the developer to provide control of the changes made to the TOE's design. **O.VULNERABILITY_ANALYSIS_-TEST** ensures that the design of the TOE is analyzed for design flaws. **O.DOCUMENTED_DESIGN** ensures that the design of the TOE is documented, permitting detailed review by evaluators and validators.

Threat/Policy	Objectives Addressing the Threat and Policies	Rationale
T.POOR_IMPLEMENTATION: Unintentional errors in implementation of the TOE design may occur, leading to flaws that may be exploited by a casually mischievous user or program.	**O.CONFIGURATION_IDENTIFICATION**: The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified, corrected with the TOE being redistributed promptly., **O.PARTIAL_FUNCTIONAL_TESTING**: The TOE will undergo some security functional testing that demonstrates the TSF satisfies some of its security functional requirements. **O.VULNERABILITY_ANALYSIS**: The TOE will undergo some vulnerability analysis demonstrate the design and implementation of the TOE does not contain any obvious flaws.	**O.CHANGE_MANAGEMENT** plays a role in countering this threat by requiring the developer to provide control of the changes made to the TOE's design. Although the previous three objectives help minimize the introduction of errors into the implementation, **O.PARTIAL_FUNCTIONAL_TEST-ING** increases the likelihood that any errors that do exist in the implementation (with respect to the functional specification, high level, and low-level design) will be discovered through testing. **O.VULNERABILITY_ANALYSIS_-TEST** helps reduce errors in the implementation that may not be discovered during functional testing. Ambiguous design documentation, and the fact that exhaustive testing of the external interfaces is not required may leave bugs in the implementation undiscovered in functional testing

Threat/Policy	Objectives Addressing the Threat and Policies	Rationale
T.POOR_TEST: Lack of or insufficient tests to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may result in incorrect TOE behavior being undiscovered thereby causing potential security vulnerabilities.	**O.DOCUMENTED_DESIGN** The design of the TOE will be adequately and accurately documented. **O.CORRECT_ TSF_OPERATION**: The TOE will provide the capability to test the TSF to ensure the correct operation of the TSF at a customer's site. **O.PARTIAL_FUNCTIONAL_TESTING**: The TOE will undergo some security functional testing that demonstrates the TSF satisfies the security functional requirements. **O.VULNERABILITY_ANALYSIS**: The TOE will undergo some vulnerability analysis demonstrate the design and implementation of the TOE does not contain any obvious flaws.	**O.DOCUMENTED_DESIGN** helps to ensure that the TOE's documented design satisfies the security functional requirements. In order to ensure the TOE's design is correctly realized in its implementation, the appropriate level of functional testing of the TOE's security mechanisms must be performed during the evaluation of the TOE. **O.PARTIAL_FUNCTIONAL_TEST-ING** increases the likelihood that any errors that do exist in the implementation (with respect to the functional specification, high level, and low-level design) will be discovered through testing. **O.VULNERABILITY_ANALYSIS_-TEST** addresses this concern by requiring a vulnerability analysis be performed in conjunction with testing that goes beyond functional testing. This objective provides a measure of confidence that the TOE does not contain security flaws that may not be identified through functional testing. While these testing activities are a necessary activity for successful completion of an evaluation, this testing activity does not address the concern that the TOE continues to operate correctly and enforce its security policies once it has been fielded. Some level of testing must be available to end users to ensure the TOE's security mechanisms continue to operate correctly once the TOE is fielded **O.CORRECT_ TSF_OPERATION** ensures that once the TOE is installed at a customer's location, the capability exists that the integrity of the TSF (hardware and software) can be demonstrated, and thus providing end users the confidence that the TOE's security policies continue to be enforced.
T.RESIDUAL_DATA: A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another.	**O.RESIDUAL_INFORMATION**: The TOE will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated.	**O.RESIDUAL_INFORMATION** counters this threat by ensuring that TSF data and user data is not persistent when resources are released by one user/process and allocated to another user/process.

Threat/Policy	Objectives Addressing the Threat and Policies	Rationale
T.TSF_COMPROMISE: A user or process may cause, through an unsophisticated attack,, TSF data, or executable code to be inappropriately accessed (viewed, modified, or deleted).	**O.RESIDUAL_INFORMATION**: The TOE will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated. **O.MANAGE**: The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.	**O.RESIDUAL_INFORMATION** is necessary to mitigate this threat, because even if the security mechanisms do not allow a user to explicitly view TSF data, if TSF data were to inappropriately reside in a resource that was made available to a user, that user would be able to inappropriately view the TSF data. **O.MANAGE** is necessary because an access control policy is not specified to control access to TSF data. This objective is used to dictate who is able to view and modify TSF data, as well as the behavior of TSF functions.
T.UNATTENDED_ SESSION: A user may gain unauthorized access to an unattended session.	**O.TOE_ACCESS**: The TOE will provide mechanisms that control a user's logical access to the TOE.	**O.TOE_ACCESS** helps to mitigate this threat by including mechanisms that place controls on user's sessions. Local administrator's sessions are locked and remote sessions are dropped after a Security Administrator defined time period of inactivity. Locking the local administrator's session reduces the opportunity of someone gaining unauthorized access the session when the console is unattended. Dropping the connection of a remote session (after the specified time period) reduces the risk of someone accessing the remote machine where the session was established, thus gaining unauthorized access to the session

Threat/Policy	Objectives Addressing the Threat and Policies	Rationale
T.UNAUTHORIZED_ ACCESS: A user may gain access to user data for which they are not authorized according to the TOE security policy.	**O.MEDIATE**: The TOE must protect user data in accordance with its security policy.	**O.MEDIATE** ensures that all accesses to user data are subject to mediation, unless said data has been specifically identified as public data**.** The TOE requires successful authentication to the TOE prior to gaining access to any controlled-access content. By implementing strong authentication to gain access to these services, an attacker's opportunity to successfully conduct a man-in-the-middle and/or password guessing attack is greatly reduced. Lastly, the TSF will ensure that all configured enforcement functions (authentication, access control rules, etc.) must be invoked prior to allowing a user to gain access to TOE or TOE mediated services. The TOE restricts the ability to modify the security attributes associated with access control rules, access to authenticated and unauthenticated services, etc to the Security Administrator. This feature ensures that no other user can modify the information flow policy to bypass the intended TOE security policy.
T.UNIDENTIFIED_ACTIONS: The administrator may not have the ability to notice potential security violations, thus limiting the administrator's ability to identify and take action against a possible security breach.	**O.AUDIT_REVIEW**: The TOE will provide the capability to selectively view audit information. **O.AUDIT_GENERATION** The TOE will provide the capability to detect and create records of security relevant events associated with users. **O.TIME_STAMPS** The TOE shall provide reliable time stamps for accountability and protocol purposes.	**O.AUDIT_REVIEW** helps to mitigate this threat by providing the Security Administrator with a required minimum set of configurable audit events that could indicate a potential security violation. By configuring these auditable events, the TOE monitors the occurrences of these events (e.g. set number of authentication failures, set number of information policy flow failures, self-test failures, etc.). **O.AUDIT_GENERATION** helps to mitigate this threat by recording actions for later review. **O.TIME_STAMPS** helps to mitigate this threat by ensuring that audit records have correct timestamps.
T.SERVER_DISRUPT Disruption of power, interface failures, or software failures that result in the TOE faulting to an unsecured state.	**O.SAFE_RECOVERY** The TSF will provide the ability to recover to a secure state.	**O.SAFE_RECOVERY** addresses this threat by providing the ability for the TOE to recover to a secure state.

Threat/Policy	Objectives Addressing the Threat and Policies	Rationale
P.ACCESS_BANNER: The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the system.	**O.DISPLAY_BANNER**: The TOE will display an advisory warning regarding use of the TOE.	**O.DISPLAY_BANNER** satisfies this policy by ensuring that the TOE displays a Security Administrator configurable banner that provides all interactive users with a warning about the unauthorized use of the TOE. *Note: Access banners with respect to the access of web content must be provided as part of that content; they are out of the scope of the TOE.*
P.ACCOUNTABILITY: The authorized users of the TOE shall be held accountable for their actions within the TOE.	**O.AUDIT_GENERATION**: The TOE will provide the capability to detect and create records of security-relevant events associated with users. **O.TIME_STAMPS**: The TOE shall provide reliable time stamps and the capability for the administrator to set the time used for these time stamps. **O.TOE_ACCESS**: The TOE will provide mechanisms that control a user's logical access to the TOE.	**O.AUDIT_GENERATION** addresses this policy by providing the Security Administrator with the capability of configuring the audit mechanism to record the actions of a specific user, or review the audit trail based on the identity of the user. Additionally, the administrator's ID is recorded when any security relevant change is made to the TOE (e.g. access rule modification, start-stop of the audit mechanism, establishment of a trusted channel, etc.). **O.TIME_STAMPS** plays a role in supporting this policy by requiring the TOE to provide a reliable time stamp (configured locally by the Security Administrator or via an external NTP server). The audit mechanism is required to include the current date and time in each audit record. All audit records that include the user ID, will also include the date and time that the event occurred. **O.TOE_ACCESS** supports this policy by requiring the TOE to identify and authenticate all authorized users prior to allowing any TOE access or any TOE mediated access on behalf of those users. While the user ID of authorized users can be assured, since they are authenticated, this PP allows unauthenticated users to access the TOE and the identity is then a presumed network identifier (e.g., IP address).

Threat/Policy	Objectives Addressing the Threat and Policies	Rationale
P.CRYPTOGRAPHY: Only NIST FIPS validated cryptography (methods and implementations) are acceptable for key management (i.e.; generation, access, distribution, destruction, handling, and storage of keys) and cryptographic services (i.e.; encryption, decryption, signature, hashing, key exchange, and random number generation services).	**O.CRYPTOGRAPHY**: The TOE shall use NIST FIPS 140-2 validated cryptographic services. **O.RESIDUAL_INFORMATION**: The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.	**O.CRYPTOGRAPHY** satisfies this policy by requiring the TOE to implement NIST FIPS validated cryptographic services. These services will provide confidentiality and integrity protection of TSF data while in transit to remote parts of the TOE. **O.RESIDUAL_INFORMATION** satisfies this policy by ensuring that cryptographic data are cleared from resources that are shared between users. Keys must be zeroized according to FIPS 140-2 and the storage location for the keys must be overwritten three or more times upon the transfer of keys to another location.
P.RATINGS_MAINTENANCE: Procedures to maintain the TOE's rating must be in place, and these procedures must be implemented to maintain the TOE's rating once it is evaluated.	**O.RATINGS_MAINTENANCE**: Procedures to maintain the TOE's rating will be documented and followed.	**O.RATINGS_MAINTENANCE** satisfies this policy by ensuring that the TOE developer has procedures and mechanisms in place to maintain the evaluated rating that is ultimately awarded the TOE. The developer must provide a plan that identifies the certified version of the TOE and its life cycle process. Identifies any plans for new releases of the TOE to include a description of the changes included in the new release and a security impact analysis of implementing the new changes. Assign and identify the TOE's developer security analyst and ensure that they follow documented procedures. TOE components must be categorized by security relevance. The categorization scheme must be documented and followed for changes to the TOE.

6.2. Rationale for the Security objectives and Security Functional Requirements for the Environment

Table 6-2. Rationale for Security Objectives for the Environment

Threat/Policy/Assumption	Objectives Addressing the Threat, Policy, or Assumption	Rationale
A.NO_GENERAL_PURPOSE There are no general purpose computing or storage repository capabilities (e.g., compilers, editors, or user applications) available on the TOE, except those permitted by the web administrator.	**OE.NO_GENERAL_PURPOSE** The IT environment will provide no general purpose computing or storage repository capabilities (e.g., compilers, editors, or user applications) available on the TOE, except those permitted by the web administrator.	**OE.NO_GENERAL_PURPOSE** directly restates the assumption as an objective.

Threat/Policy/Assumption	Objectives Addressing the Threat, Policy, or Assumption	Rationale
A.NO_EVIL Administrators are non-hostile, appropriately trained, and follow all administrator guidance.	**OE.NO_EVIL** Sites using the TOE shall ensure that administrators are non-hostile, appropriately trained, and follow all administrator guidance.	**OE.NO_EVIL** directly restates the assumption as an objective.
A.PHYSICAL Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the IT environment.	**OE.PHYSICAL** Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the IT environment.	**OE.PHYSICAL** directly restates the assumption as an objective.
A.PROVIDERS_GOOD Content providers will appropriately control the visibility of their content (i.e., they will establish appropriate access controls) based on the sensitivity of that content. Content providers will also follow published guidance regarding installation of content.	**OE.PROVIDERS_GOOD** Sites using the TOE shall provide content providers with guidance on how to protect controlled access information and how to develop safe and appropriate content.	**OE.PROVIDERS_GOOD** directly addresses the assumption.
A.SYSTEM_HIGH All web users who can access the system have legal authorization for the information, although they may not have need to know.	**OE.SYSTEM_HIGH** Sites using the TOE shall ensure that all authorized users of and networks connecting to the TOE have a legal ability to see the information provided (even if they lack need to know).	**OE.SYSTEM_HIGH** directly addresses the assumption.
T.REPLAY A threat agent may replay valid identification and authentication that has been captured to disguise itself as an Authorized Administrator of the TOE.	**OE.AUTHORIZED_USERS** The IT Environment must ensure that all users using the IT Environment directly have been authorized and are accountable for their actions. **OE.AS_REMOTE_ACCESS** Administrators of the web server may access the web server directly (e.g., console port) or remotely as long as the communication path is protected (e.g., physically or through encryption).	**OE.AUTHORIZED_USERS** partially addresses this threat by ensuring that the environment provides authorization for users. **OE.AS_REMOTE ACCESS** also addresses this threat, by providing protection for the communications path for remote users, thus reducing the risk of observation and capture of credentials.
T.TCPIP_ATTACK A threat agent may take advantage of a published vulnerability against protocols layers below HTTP (e.g., TCP or IP), resulting in the TOE being unable to respond properly to valid requests.	**OE.BASIC_ROBUSTNESS** The IT environment must be sufficiently robust to protect against the casual attacker using published exploits.	**OE.BASIC_ROBUSTNESS** addresses this threat by requiring the IT environment to be protected against published exploits.

Threat/Policy/Assumption	Objectives Addressing the Threat, Policy, or Assumption	Rationale
T.UNDERLYING_PROT A threat agent may be able to obtain unauthorized access to TSF data or contents through inadequate handling of TOE requests to protect underlying data objects.	**OE.SEP_ENVIRONMENT** The IT Environment must provide sufficient mechanisms to protect the TSF's data and memory during storage and execution. **OE.ACCESS_CONTROL** The IT environment must provide the TOE with an access control mechanism suitable to protect the TSF and content provider data and configuration. **OE.BASIC_ROBUSTNESS** The IT environment must be sufficiently robust to protect against the casual attacker using published exploits.	**OE.SEP_ENVIRONMENT** helps address this threat by protecting the TSF data during TSF execution. **OE.ACCESS_CONTROL** helps address this threat by providing access control mechanisms to permit the TSF to protect its data. **OE.BASIC_ROBUSTNESS** helps address this threat by providing sufficient mechanisms to provide accountability and robustness of implementation of the IT environment.
T.VIRTUAL_ADDR_FAILURE A threat agent may be able to subvert the TOE through execution of another process on the IT platform, which modifies the operational code of the TOE.	**OE.SEP_ENVIRONMENT** The IT Environment must provide sufficient mechanisms to protect the TSF's data and memory during storage and execution. **OE.BASIC_ROBUSTNESS** The IT environment must be sufficiently robust to protect against the casual attacker using published exploits.	**OE.SEP_ENVIRONMENT** addressed this threat by protecting the TSF data and code during TSF execution. **OE.BASIC_ROBUSTNESS** helps address this threat by providing sufficient mechanisms to provide accountability and robustness of implementation of the IT environment.

The IT environment objectives are also designed to provide additional support and underlying mechanisms for the TOE Objectives. For example, **OE.RELIABLE_TIME_STAMPS** directly supports the TOE objective **O.TIME_STAMPS**, by providing the underlying mechanism.

6.3. Rationale for TOE Security Requirements

Table 6-3. Rationale for TOE Security Requirements presents a mapping between objectives for the TOE and the TOE Security Requirements that implement those objectives.

Table 6-3. Rationale for TOE Security Requirements

Objectives	Requirements Addressing the Objective	Rationale
O.ADMIN_GUIDANCE: The TOE will provide administrators with the necessary information for secure management.	ADO_DEL.1 ADO_IGS.1 ADO_ADM.1 AGD_USR.1 AVA_MSU.1	**ADO_DEL.1** ensures that the administrator is provided documentation that instructs them how to ensure the delivery of the TOE, in whole or in parts, has not been tampered with or corrupted during delivery. This requirement ensures the administrator has the ability to begin their TOE installation with a *clean* (e.g., malicious code has not been inserted once it has left the developer's control) version of the TOE, which is necessary for secure management of the TOE. **ADO_IGS.1** ensures the administrator has the information necessary to install the TOE in the evaluated configuration. Often times a vendor's product contains software that is not part of the TOE and has not been evaluated. The Installation, Generation and Startup (IGS) documentation ensures that once the administrator has followed the installation and configuration guidance the result is a TOE in a secure configuration. **AGD_ADM.1** mandates the developer provide the administrator with guidance on how to operate the TOE in a secure manner. This includes describing the interfaces the administrator uses in managing the TOE, security parameters that are configurable by the administrator, how to configure the TOE's rule set and the implications of any dependencies of individual rules. The documentation also provides a description of how to setup and review the auditing features of the TOE. **AGD_USR.1** is intended for non-administrative users, but could be used to provide guidance on security that is common to both administrators and non-administrators (e.g., password management guidelines). Since the non-administrative users of this TOE are limited to proxy users it is expected that the user guidance would discuss the secure use of proxies and how the single-use authentication mechanism is used. The use of the single-use authentication mechanism would not have to be repeated in the administrator's guide. **AVA_MSU.1** ensures that the guidance documentation is complete and consistent, and notes all requirements for external security measures.

Objectives	Requirements Addressing the Objective	Rationale
O.AUDIT_GENERATION: The TOE will provide the capability to detect and create records of security-relevant events associated with users	FAU_GEN.1-NIAP-0410 FAU_GEN.2-NIAP-0410 FAU_SEL.1-NIAP-0407 FIA_USB.1-NIAP-0351	**FAU_GEN.1-NIAP-0410** defines the set of events that the TOE must be capable of recording. This requirement ensures that the Administrator has the ability to audit any security relevant event that takes place in the TOE. This requirement also defines the information that must be contained in the audit record for each auditable event. This requirement also places a requirement on the level of detail that is recorded on any additional security functional requirements an ST author adds to this PP. **FAU_GEN.2-NIAP-0410** ensures that the audit records associate a user identity with the auditable event. In the case of authorized users, the association is accomplished with the userid. In all other cases, the association is based on the source network identifier, which is presumed to be the correct identity, but cannot be confirmed since these subjects are not authenticated. **FAU_SEL.1-NIAP-0407** allows the Security Administrator to configure which auditable events will be recorded in the audit trail. This provides the administrator with the flexibility in recording only those events that are deemed necessary by site policy, thus reducing the amount of resources consumed by the audit mechanism. **FIA_USB.1-NIAP-0351** plays a role is satisfying this objective by requiring a binding of security attributes associated with users that are authenticated with the subjects that represent them in the TOE. This only applies to authorized users, since the identity of unauthenticated users cannot be confirmed. Therefore, the audit trail may not always have the proper identity of the subject that causes an audit record to be generated (e.g., presumed network address of an unauthenticated user may be a spoofed address).
O.AUDIT_PROTECTION: The TOE will provide the capability to protect audit information.	FAU_SAR.2 FAU_STG.1-NIAP-0429 FAU_STG.3 FAU_STG.NIAP-0414-1-NIAP-0429 FMT_MOF.1	**FAU_SAR.2** restricts the ability to read the audit trail to the Audit Administrator, thus preventing the disclosure of the audit data to any other user. However, the TOE is not expected to prevent the disclosure of audit data if it has been archived or saved in another form (e.g., moved or copied to an ordinary file). The **FAU_STG** family dictates how the audit trail is protected. FAU_STG.1-NIAP-0429 restricts the ability to delete audit records to the Security Administrator. FAU_STG.3 requires the TOE to alert the administrator when the audit trail becomes full, and FAU_STG.NIAP-0414-1-0429, defines the actions that must be available to the administrator, as well as the action to be taken if there is no response. This helps to ensure that audit records are kept until the Security Administrator deems they are no longer necessary. This requirement also ensures that no one has the ability to modify audit records (e.g., edit any of the

Objectives	Requirements Addressing the Objective	Rationale
		information contained in an audit record). This ensures the integrity of the audit trail is maintained. **FMT_MOF.1** restricts the capability to modify the behavior of the audit and alarm functions to the Security Administrator. While the Audit Administrator has the capability to choose how they will review the audit trail, they do not have the capability to select what events are audited. This requirement ensures that only the Security Administrator can turn audit on or off, this ensuring users actions are audited according to a site defined policy.
O.AUDIT_REVIEW: The TOE will provide the capability to selectively view audit information,.	FAU_SAR.1 FAU_SAR.3	**FAU_SAR.1** provides the Audit Administrator with the capability to read all the audit data contained in the audit trail. This requirement also mandates the audit information be presented in a manner that is suitable for the Audit Administrator to interpret the audit trail, which is subject to interpretation. It is expected that the audit information be presented in such a way that the Audit Administrator can examine an audit record and have the appropriate information (that required by FAU_GEN.2) presented together to facilitate the analysis of the audit review. **FAU_SAR.3** complements FAU_SAR.1 by providing the Audit Administrator the flexibility to specify criteria that can be used to search or sort the audit records residing in the audit trail. FAU_SAR.3 requires the Audit Administrator be able to establish the audit review criteria based on a userid and source subject identity, so that the actions of a user can be readily identified and analyzed. The criteria also includes a destination subject identity so the Audit Administrator can determine what network traffic is destined for an individual machine. Allowing the Audit Administrator to perform searches or sort the audit records based on dates, times, subject identities, destination service identifier, or transport layer protocol provides the capability to extract the network activity to what is pertinent at that time in order facilitate the Audit Administrator's review. Being able to search on the destination service identifier affords the Audit Administrator the opportunity to see what traffic is destined for a service (e.g., TCP port) or set of services regardless of where the traffic originated. It is important to note that the intent of sorting in this requirement is to allow the Audit Administrator the capability to organize or group the records associated with a given criteria. For example, if the Audit Administrator wanted to see what network traffic was destined for the set of TCP ports 1-1024, they would be able to have the audit data presented in such a way that all the traffic for TCP port 1 was grouped together, all the traffic for port 2 was grouped together and so on.

Objectives	Requirements Addressing the Objective	Rationale
O.CONFIGURATION_IDENTIFICATION: The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified, corrected with the TOE being redistributed promptly.	ACM_CAP.2 ALC_FLR.2	**ACM_CAP.2** addresses this objective by requiring that that there be a unique reference for the TOE, and that the TOE is labeled with that reference. It also requires that there be a CM system in place, and that the configuration items that comprise the TOE by uniquely identified. This provides a clear identification of the composition of the TOE. **ALC_FLR.2** addresses this objective by requiring that there be a mechanism in place for identifying flaws subsequent to fielding, and for distributing those flaws to entities operating the system.
O.CORRECT_ TSF_OPERATION: The TOE will provide the capability to test the TSF to ensure the correct operation of the TSF at a customer's site.	FPT_AMT.1 FPT_TST.1/CR FPT_TST.1/NC FPT_TST_EXP.1/KG	The **FPT_AMT.1** component verifies the correct operation of the underlying abstract machine (if there is one) to ensure that the required IT environment properties are provided. The **FPT_TST** components address this objective by providing the capability to test various security critical aspects of the TOE's operation. **FPT_TST.1/CR** and **FPT_TST_EXP.1/KG** serve to provide assurance that the cryptographic mechanisms in the TOE are operating correctly. **FPT_TST.1/NC** provides assurance that the non-cryptographic mechanisms are working correctly. The specific cryptographic tests address the critical nature and specific handling of the cryptographic related TSF data. Since the cryptographic TSF data has specific FIPS PUB requirements associated with them it is important to ensure that any fielded testing on the integrity of these data maintains the same level of scrutiny as specified in the FCS functional requirements. The explicit cryptographic test requirement allows the Security Administrator the option of having the cryptographic self-tests executed after the generation of every key. This may not be practical for some installations, therefore it is left to the Security Administrator's discretion.
O.CRYPTOGRAPHY_VALIDATED: The TOE will use NIST FIPS 140-2 validated cryptomodules for cryptographic services implementing NIST-approved security functions and random number generation services used by cryptographic functions.	FCS_BCM_EXP.1 FCS_CKM.1 FCS_CKM.4 FCS_CKM_EXP.1 FCS_COP_EXP.1 FCS_COP.1 (2) FCS_COP.1 (3)	The FCS requirements used in this PP satisfy this objective by levying requirements that ensure the cryptographic standards include the NIST FIPS publications (where possible) and NIST approved ANSI standards. The intent is to have the satisfaction of the cryptographic standards be validated through a NIST FIPS 140 validation. **FCS_BCM_EXP.1** is an explicit component that specifies what NIST FIPS rating level the cryptographic module must satisfy. The level specifies the degree of testing of the module. The higher the level, the more extensive the module is tested. **FCS_CKM.1** defines the specific key sizes and standards that are required for the generation of symmetric and asymmetric keys. **FCS_CKM.4** defines the mechanism to be used

Objectives	Requirements Addressing the Objective	Rationale
		for key zeroization. **FCS_CKM_EXP.1** defines the method of key establishment for a wide variety of key establishment techniques. **FCS_CKM_EXP.2** defines the method of random number generation that is used by the cryptographic functionality, and requires that the generator be FIPS approved. **FCS_COP.1 (2)** defines the method of digital signature verification and generation. **FCS_COP.1 (3)** defines the mechanism to be used for Cryptographic Hashing, and requires that it be a NIST-approved cryptomodule.
O.CRYPTOGRAPHIC_FUNCTIONS: The TOE will provide cryptographic functions for its own use, including encryption/decryption and digital signature operations.	FCS_CKM.1 FCS_CKM.4 FCS_CKM_EXP.1 FCS_COP_EXP.1 FCS_COP.1 (2) FCS_COP.1 (3)	The FCS requirements used in this PP satisfy this objective by levying requirements that ensure that appropriate cryptographic functions are made available. **FCS_CKM.1** defines the specific key sizes and standards that are required for the generation of symmetric and asymmetric keys. **FCS_CKM.4** defines the mechanism to be used for key zeroization. **FCS_CKM_EXP.1** defines the method of key establishment for a wide variety of key establishment techniques. **FCS_CKM_EXP.2** defines the method of random number generation that is used by the cryptographic functionality. **FCS_COP.1 (2)** defines the method of digital signature verification and generation. **FCS_COP.1 (3)** defines the mechanism to be used for Cryptographic Hashing.
O.DISPLAY_BANNER: The TOE will display an advisory warning regarding use of the TOE.	FTA_TAB.1	**FTA_TAB.1** meets this objective by requiring the TOE display a Security Administrator defined banner before a user can establish an authenticated session. This banner is under complete control of the Security Administrator in which they specify any warnings regarding unauthorized use of the TOE and remove any product or version information if they desire.
O.DOCUMENTED_DESIGN: The design of the TOE is adequately and accurately documented.	ADV_FSP.1 ADV_HLD.1 ADV_RCR.1	**ADV_FSP.1** requires that the interfaces to the TOE be documented and specified. **ADV_HLD.1** requires that the high level design of the TOE be documented and specified and that said design be shown to correspond to the

Objectives	Requirements Addressing the Objective	Rationale
		interfaces. **ADV_RCR.1** requires that there be a correspondence between adjacent layers of the design decomposition.
O.MANAGE: The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.	FMT_MOF.1 FMT_MSA.1 FMT_MSA.2 FMT_MSA.3-NIAP-0429 FMT_MTD.1 FMT_REV.1 FMT_SMR.1	**FMT_MOF.1** requires that the ability to use particular TOE capabilities be restricted to the Administrator. **FMT_MSA.1** requires that the ability to perform operations on security attributes be restricted to particular roles. **FMT_MSA.2** provides the Security Administrator the capability to manipulate the security attributes to facilitate the construction of the rule set. An example of this would be to group a set of service identifiers that are to have the same rule applied, rather than having to specify a separate rule for each service identifier. **FMT_MSA.3-NIAP-0429** requires that default values used for security attributes are restrictive, and that the Administrator has the ability to override those values. **FMT_MTD.1** requires that the ability to manipulate TOE content is restricted to Administrators and authorized Content Providers. **FMT_REV.1** restricts the ability to revoke attributes to the administrator. **FMT_SMR.1** defines the specific security roles to be supported.

Objectives	Requirements Addressing the Objective	Rationale
O.MEDIATE: The TOE must protect user data in accordance with its security policy.	FDP_ACC.2/WU FDP_ACF.1-NIAP-0407/WU FDP_UCT.1/WU FDP_UIT.1/WU FDP_ACC.2/CP FDP_ACF.1-NIAP-0407/CP FMT_REV.1 FPT_RVM.1	Compliant TOEs have two security policies: one for web users, and one for content providers. The **/WU** iterations serve to define the SFP for Web Users. The basic policy is defined by **FDP_ACC.2/WU** and **FDP_ACF.1-NIAP-0407/WU**, which defines the subjects, objects, operations, and attributes controlled by the policy, together with the policy rules. For controlled-access data, these are augmented by **FDP_UIT.1/WU** and **FDB_UCT.1/WU**, which protect the data during transit. **FTP_ITC.1** serves to protect controlled access content during transmission. The **/CP** iterations serve to define the SFP for Content Providers. The basic policy is defined by **FDP_ACC.2/CP** and **FDP_ACF.1-NIAP-0407/CP**, which defines the subjects, objects, operations, and attributes controlled by the policy, together with the policy rules. **FMT_REV.1** is a management requirement that affords the Security Administrator the ability to immediately revoke access under an SFP. **FPT_RVM.1** ensures that all operations in the TOE are subject to the security policy; there is no mechanism provided to bypass the policy.
O.PARTIAL_FUNCTIONAL_TESTING: The TOE will undergo some security functional testing that demonstrates the TSF satisfies some of its security functional requirements.	ATE_COV.1 ATE_FUN.1 ATE_IND.2	**ATE_FUN.1** requires that developer provide test documentation for the TOE, including test plans, test procedure descriptions, expected test results, and actual test results. These need to identify the functions tested, the tests performed, and test scenarios. They require that the developer run those tests, and show that the expected results were achieved. **ATE_COV.1** requires that there be a correspondence between the tests in the test documentation and the TSF as described in the functional specification. **ATE_IND.2** requires that the evaluators test a subset of the TSF to confirm correct operation, on an equivalent set of resources to those used by the developer for testing. These sets should include a subset of the developer run tests.
O.RATINGS_MAINTENANCE: Procedures to maintain the TOE's rating will be documented and followed.	AMA_AMP.1 AMA_CAT.1 AMA_EVD.1 AMA_SIA.1	The **AMA** family of requirements is incorporated into this PP to ensure the TOE developer has procedures and mechanisms in place to maintain the evaluated rating that is ultimately awarded the TOE. These requirements are somewhat related to the ACM family of requirements in that changes to the TOE and its evidence must be managed, but the AMA requirements ensure the appropriate level of analysis is performed on any changes

Objectives	Requirements Addressing the Objective	Rationale
		made to the TOE to ensure the changes do not affect the TOE's ability to enforce its security policies. **AMA_AMP.1** requires the developer to develop an assurance maintenance (AM) plan that describes how the assurance gained from an evaluation will be maintained, and that any changes to the TOE will be analyzed to determine the security impact, if any, of the changes that are made. This requirement mandates the developer assign personnel to fulfill the role of a security analyst that is responsible for ensuring the changes made to the TOE will not adversely impact the TOE and that it will continue to maintain its evaluation rating. **AMA_CAT.1** is used to focus the security analyst's scope in analyzing the changes made to the TOE. Components of the TOE are categorized according to the components security relevance in the TOE. For example, a TOE that conforms to this PP might have a component such as a scheduler that is deemed to play no role in satisfying the security requirements and therefore would not get a lot of attention from the security analyst. On the other hand, the network stack plays an important role in satisfying the FDP_IFF requirements, and others, and would require a great deal of scrutiny by the analyst. **AMA_EVD.1** ensures that the developer is following the AM plan by requiring the developer to provide evidence. This is an important component in assuring that the procedures required by AMA_AMP.1 are pertinent to the maintenance of the TOE's rating. **AMA_SIA.1** plays an important role in satisfying this objective by requiring the developer's security analyst to document any modifications (or additions) to the TOE that affect the enforcement of the TOE's security policies. Additionally, the evidence required documents the analysis performed by the analyst and provides a degree of confidence that the appropriate level of analysis was performed and the continued evaluation rating of the new version of the TOE is warranted.
O.RESIDUAL_ INFORMATION: The TOE will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated.	FDP_RIP.2 FCS_CKM.4	**FDP_RIP.2** is used to ensure the contents of resources are not available to subjects other than those explicitly granted access to the data. **FCS_CKM.4** applies to the destruction of cryptographic keys used by the TSF. This requirement specifies how and when cryptographic keys must be destroyed. The proper destruction of these keys is critical in ensuring the content of these keys cannot possibly be disclosed when a resource is reallocated to a user.

Objectives	Requirements Addressing the Objective	Rationale
O.PARTIAL_SELF_PROTECTION: The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure through its own interfaces.	FPT_SEP_EXP.1 FPT_RVM.1	The explicitly specific component **FPT_SEP_EXP.1** was chosen to ensure the TSF provides a domain that protects itself from untrusted users. If the TSF cannot protect itself it cannot be relied upon to enforce its security policies. The explicitly specified version was used to distinguish the aspects of FPT_SEP provided by the TOE vs. the aspects provided by the IT environment. The inclusion of **FPT_RVM.1** ensures that the TSF makes policy decisions on all interfaces that perform operations on subjects and objects that are scoped by the policies. Without this non-bypassability requirement, the TSF could not be relied upon to completely enforce the security policies, since an interface(s) may otherwise exist that would provide a user with access to TOE resources (including TSF data and executable code) regardless of the defined policies. This includes controlling the accessibility to interfaces, as well as what access control is provided within the interfaces.
O.SAFE_RECOVERY The TSF will provide the ability to recover to a secure state.	FPT_RCV.2	**FPT_RCV.2** requires that the TSF provide the capability to return to a secure state in an automatic fashion.
O.TIME_STAMPS The TOE will provide reliable time stamps for accountability and protocol purposes.	FPT_STM.1	**FPT_STM.1** requires that the TSF provide time stamps for its own use.
O.TOE_ACCESS: The TOE will provide mechanisms that control a user's logical access to the TOE.	FIA_AFL.1-NIAP-0425 FIA_ATD.1 FIA_UID.1 FIA_UAU.1 FIA_UAU.7 FTA_SSL.1 FTA_SSL.2 FTA_SSL.3/IN FTA_SSL.3/WU AVA_SOF.1	**FIA_AFL.1-NIAP-0425** provides a detection mechanism for unsuccessful authentication attempts by remote administrators, authenticated proxy users and authorized IT entities. The requirement enables a Security Administrator settable threshold that prevents unauthorized users from gaining access to authorized user's account by guessing authentication data by locking the targeted account. Thus, limiting an unauthorized user's ability to gain unauthorized access to the TOE. **FIA_ATD.1** defines the attributes of users, including a userid that is used to by the TOE to determine a user's identity and enforce what type of access the user has to the TOE (e.g., the TOE associates a userid with any role(s) they may assume). **FIA_UID.1** requires that a user be identified to the TOE in order to access anything other than public content. **FIA_UAU.1** requires that a user be authenticated by the TOE before accessing anything other than public content.

Objectives	Requirements Addressing the Objective	Rationale
		FIA_UAU.7 provides that the authentication data provided by the user is not echoed back in plaintext, thus serving to protect that data. The **FTA_SSL** components all deal with automatic session locking and termination, either initiated by the TSF (**FTA_SSL.1, FTA_SSL.3/IN**), a user (**FTA_SSL.2**), or a web user (**FTA_SSL.3/WU**). The **AVA_SOF.1** requirement is applied to the password mechanism used by the local administrator (The single use authentication mechanism supplied by the IT environment (i.e., authentication server) has this same assurance requirement levied against it to ensure a consistent level of assurance.) For this TOE, the strength of function specified is medium. This requirement ensures the developer has performed an analysis of the password mechanism to ensure the probability of guessing a local administrator's password would require a high-attack potential, as defined in Annex B of the CEM. This analysis takes into account the password space, as well as any feature of the password mechanism that plays a role in limiting the number of failed authentication attempts within a given time period.
O.USER_CONFIDENCE The TOE will provide mechanisms that permit web users to have confidence that received controlled-access data comes from the TOE.	FCO_NRO.2	**FCO_NRO.2** requires that the TOE provide mechanisms that provide the web user with assurance that the originator of the information actually sent that information.
O.VULNERABILITY_ANALYSIS: The TOE will undergo some vulnerability analysis demonstrate the design and implementation of the TOE does not contain any obvious flaws.	AVA_VLA.1	The **AVA_VLA.1** component provides the necessary level of confidence that vulnerabilities do not exist in the TOE that could cause the security policies to be violated. AVA_VLA.1 requires the developer to perform a systematic search for potential vulnerabilities in all the TOE deliverables. For those vulnerabilities that are not eliminated, a rationale must be provided that describes why these vulnerabilities cannot be exploited by a threat agent with a moderate attack potential, which is in keeping with the desired assurance level of this TOE. As with the functional testing, a key element in this component is that an independent assessment of the completeness of the developer's analysis is made, and more importantly, an independent vulnerability analysis coupled with testing of the TOE is performed. This component provides the confidence that security flaws do not exist in the TOE that could be exploited by a threat agent of moderate (or lower) attack potential to violate the TOE's security policies.

6.4. Assurance Security Requirements Rationale

EAL2 was chosen because it was the lowest EAL at which all of the assurance requirements necessary to meet the objectives were present. These requirements have been detailed in Table

6-3. Rationale for TOE Security Requirements. It was also felt that EAL2 provided sufficient assurance for correct operation of the server functional elements, while still allowing servers developed as commercial products to be acceptable.

6.5. Dependency Requirements Rationale

Table 6-4 identifies the dependencies allocated to the functional requirements addressed in this protection profile. Table 6-5 identifies the dependencies allocated to the assurance requirements addressed within this protection profile.

Table 6-4. Functional Requirement Dependencies

Requirement	Dependency	Dependency Analysis and Rationale
FAU_GEN.1-NIAP-0410	FPT_STM.1	Satisfied
FAU_GEN.2-NIAP-0410	FAU_GEN.1 FIA_UID.1	Satisfied by FAU_GEN.1-NIAP-0410 Satisfied
FAU_SAR.1	FAU_GEN.1	Satisfied by FAU_GEN.1-NIAP-0410
FAU_SAR.2	FAU_SAR.1	Satisfied
FAU_SAR.3	FAU_SAR.1	Satisfied
FAU_SEL.1-NAIP-0407	FAU_GEN.1 FMT_MTD.1	Satisfied by FAU_GEN.1-NIAP-0410 Satisfied
FAU_STG.1-NIAP-0429	FAU_GEN.1	Satisfied by FAU_GEN.1-NIAP-0410
FAU_STG.NIAP-0414-1-NIAP-0429	FAU_STG.1 FMT_MTD.1	Satisfied by FAU_STG.1-NIAP-0429 Satisfied
FAU_STG.3	FAU_STG.1	Satisfied by FAU_STG.1-NIAP-0429
FCO_NRO.2	FIA_UID.1	Satisfied
FCS_BCM_EXP.1	No Dependencies Specified	
FCS_CKM.1	FCS_COP.2 or FCS_COP.1 FCS_CKM.4 FMT_MSA.2	Satisfied by FCS_COP.1 (1), a variant of FCS_COP.1 specifically for FIPS Compliance Satisfied Satisfied
FCS_CKM_EXP.1	No Dependencies Specified	
FCS_CKM.4	FDP_ITC.1 or FCS_CKM.1 FMT_MSA.2	Satisfied by FCS_CKM.1 Satisfied
FCS_COP.1 (1)	No Dependencies Specified	
FCS_COP_EXP.1	No Dependencies Specified	
FCS_COP.1 (2)	No Dependencies Specified	
FCS_COP.1 (3)	No Dependencies Specified	
FDP_ACC.2/WU	FDP_ACF.1	Satisfied by FDP_ACC.1/WU

Requirement	Dependency	Dependency Analysis and Rationale
FDP_ACF.1/WU	FDP_ACC.1	Satisfied by FDP_ACC.2/WU, which is hierarchical to FDP_ACC.1
	FMT_MSA.3	Satisfied by FMT_MSA.3-NIAP-0492
FDP_UCT.1/WU	FTP_ITC.1 or FTP_TRP.1	Satisfied by FTP_ITC.1
	FDP_ACC.1 or FDP_IFC.1	Satisfied by FDP_ACC.1/WU
FDP_UIT.1/WU	FTP_ITC.1 or FTP_TRP.1	Satisfied by FTP_ITC.1
	FDP_ACC.1 or FDP_IFC.1	Satisfied by FDP_ACC.1/WU
FDP_ACC.2/CP	FDP_ACF.1	Satisfied by FDP_ACC.1/CP
FDP_ACF.1/CP	FDP_ACC.1	Satisfied by FDP_ACC.2/CP, which is hierarchical to FDP_ACC.1
	FMT_MSA.3	Satisfied by FMT_MSA.3-NIAP-0492
FDP_RIP.2	No Dependencies	
FIA_AFL.1-NIAP-0425	FIA_UAU.1	Satisfied
FIA_ATD.1	No Dependencies	
FIA_UAU.1	FIA_UID.1	Satisfied
FIA_UAU.7	FIA_UAU.1	Satisfied
FIA_UID.1	No Dependencies	
FIA_USB.1-NIAP-0351	FIA_ATD.1	Satisfied
FMT_MOF.1	FMT_SMR.1	Satisfied
FMT_MSA.1	FMT_SMR.1	Satisfied
FMT_MSA.2	FMT_MSA.1	Satisfied
	FMT_SMR.1	Satisfied
	ADV_SPM.1	ADV_SPM.1 only comes in at EAL4. ADV_SPM.1 was included in order to provide the definition of what is a secure value for an attribute. This information will be provided in other evaluation documentation.
	FDP_ACC.1 or FDP_IFC.1	Satisfied by FDP_ACC.1/WU and FDP_ACC.1/CP
FMT_MSA.3	FMT_MSA.1	Satisfied
	FMT_SMR.1	Satisfied
FMT_MTD.1	FMT_SMR.1	Satisfied
FMT_REV.1	FMT_SMR.1	Satisfied
FMT_SMR.1	FIA_UID.1	Satisfied

FPT_AMT.1	No Dependencies	
FPT_RCV.2	FPT_TST.1 AGD_ADM.1 ADV_SPM.1	Satisfied by FPT_TST.1/CR and FPT_TST.1/NC Satisfied ADV_SPM.1 only comes in at EAL4. ADV_SPM.1 was included in order to provide the definition of what is a secure state. This information will be provided in other evaluation documentation.
FPT_RVM.1	No Dependencies	
FPT_SEP_EXP.1	No Dependencies Specified	
FPT_STM.1	No Dependencies	
FPT_TST.1/CR	FPT_AMT.1	Satisfied
FPT_TST.1/NC	FPT_AMT.1	Satisfied
FPT_TST_EXP.1/KG	No Dependencies Specified	
FTA_SSL.1	FIA_UAU.1	Satisfied
FTA_SSL.2	FIA_UAU.1	Satisfied
FTA_SSL.3/IN	No Dependencies	
FTA_SSL.3/WU	No Dependencies	
FTA_TAB.1	No Dependencies	
FTP_ITC.1	No Dependencies	

Table 6-5. Assurance Requirement Dependencies

Requirement	Dependency	Dependency Analysis and Rationale
ACM_CAP.2	No Dependencies	
ADO_DEL.1	No Dependencies	
ADO_IGS.1	AGD_ADM.1	Satisfied
ADV_FSP.1	ADV_RCR.1	Satisfied
ADV_HLD.1	ADV_FSP.1 ADV_RCR.1	Satisfied Satisfied
ADV_RCR.1	No Dependencies	
AGD_ADM.1	ADV_FSP.1	Satisfied
AGD_USR.1	ADV_FSP.1	Satisfied
ALC_FLR.2	No Dependencies	
ATE_COV.2	ADV_FSP.1	Satisfied

Requirement	Dependency	Dependency Analysis and Rationale
	ATE_FUN.1	Satisfied
ATE_FUN.1	No Dependencies	
ATE_IND.2	ADV_FSP.1 AGD_ADM.1 AGD_USR.1 ATE_FUN.1	Satisfied Satisfied Satisfied Satisfied
AVA_MSU.1	ADO_IGS.1 ADV_FSP.1 AGD_ADM.1 AGD_USR.1	Satisfied Satisfied Satisfied Satisfied
AVA_SOF.1	ADV_FSP.1 ADV_HLD.1	Satisfied Satisfied
AVA_VLA.1	ADV_FSP.1 ADV_HLD.1 AGD_ADM.1 AGD_USR.1	Satisfied Satisfied Satisfied Satisfied
AMA_AMP.1	ACM_CAP.2 ALC_FLR.1 AMA_CAT.1	Satisfied Satisfied Satisfied
AMA_CAT.1	ACM_CAP.2	Satisfied
AMA_EVD.1	AMA_AMP.1 AMA_SIA.1	Satisfied Satisfied
AMA_SIA.1	AMA_CAT.1	Satisfied

6.6. Rationale for Not Satisfying All Dependencies

Each functional requirement, including explicit requirements was analyzed to determine that all dependencies were satisfied. All requirements were then analyzed to determine that no additional dependencies were introduced as a result of completing each operation. Table 6-4. Functional Requirement Dependencies identifies the functional requirement, its correspondent dependency and the analysis and rationale for not supporting the dependency in this PP.

6.7. Rationale for Explicit requirements

Table 6-6 presents the rationale for the inclusion of the explicit requirements found in this PP.

Table 6-6. Rationale For The Inclusion Of The Explicit Requirements

Explicit Requirement	Identifier	Rationale
FCS_BCM_EXP.1	Baseline Cryptographic Module	The CC does not provide a means to specify the use of NIST FIPS validated cryptography as a baseline for the cryptographic module. FCS_BCM_EXP.1 is an explicit component that specifies what NIST FIPS rating level the cryptographic module must satisfy. The level specifies the degree of testing of the module.
FCS_CKM_EXP.1	Cryptographic Key Establishment	The CC does not provide a means to specify the use of NIST FIPS validated cryptography as a baseline for the cryptographic module. FCS_CKM_EXP.1 is an explicit component that specifies only those methods that work within FIPS approved approaches.
FCS_COP_EXP.1	Random number generation	The CC does not provide a means to specify the use of NIST FIPS validated cryptography as a baseline for the cryptographic module. FCS_COP_EXP.1 is an explicit component that specifies use of a FIPS approved random number generator.
FCS_COP.1 (2)	Cryptographic Operation (Digital Signature Generation/Verification)	The CC does not provide a means to specify the use of NIST FIPS validated cryptography as a baseline for the cryptographic module. FCS_COP.1 (2) is an explicit component that specifies use of a FIPS approved digital signature generation/verification approach.
FCS_COP.1 (3)	Cryptographic Operation (Cryptographic Hashing Function)	The CC does not provide a means to specify the use of NIST FIPS validated cryptography as a baseline for the cryptographic module. FCS_COP.1 (3) is an explicit component that specifies use of a FIPS approved cryptographic hashing function.
FPT_SEP_EXP.1	Application Domain Separation	The CC does not provide a means of specifying domain separation in a manner that is truly applicable to applications running on an OS. This explicit version of FPT_SEP captures just the application aspects of domain separation.
FPT_TST_EXP1.1	TSF testing (Key Generation Components)	Key generation requires specific testing techniques under FIPS. This explicit component captures those techniques.

Explicit Requirement	Identifier	Rationale
FIT_PPC_EXP.1	IT Environment Profile Compliance	The security of the TOE depends on appropriate security being provided by the underlying platform. This could have been specified by detailing each of the CAPP or OS PP requirements, which would lengthen the ST and potentially confuse the end users. To simplify use of the PP, an explicit requirement was created for IT environment compliance with an appropriate PP.

7.0. REFERENCES

1) Common Criteria for Information Technology Security Evaluation*, CCIB-98-031 Version 2.1, August 1999.*

2) *Department of Defense Chief Information Officer Guidance and Policy Memorandum No. 6-8510,* Guidance and Policy for the Department of Defense Global Information Grid Information Assurance (GIG), *June 2000.*

3) U.S. Government Traffic-Filter Firewall Protection Profile for Medium Robustness Environments, *Version 1.4, May 1, 2000.*

4) U.S. Department of Defense Application-Level Firewall Protection Profile for Medium Robustness Environments, *Version 1.1, December 2001.*

5) Information Assurance Technical Framework, *Version 3.0, September 2000.*

6) *Federal Information Processing Standard Publication (FIPS-PUB) 46-3,* Data Encryption Standard (DES), *October 1999.*

7) *Federal Information Processing Standard Publication (FIPS-PUB) 140-2,* Security Requirements for Cryptographic Modules, *May 25, 2001.*

8) *Internet Engineering Task Force,* IP Encapsulating Security Payload (ESP), *RFC 2406, November 1998.*

9) *Internet Engineering Task Force,* Internet Key Exchange (IKE), *RFC 2409, November 1998.*

10) *Internet Engineering Task Force,* ESP CBC-Mode Cipher Algorithms, *RFC 2451, November 1998.*

11) *Internet Engineering Task Force,* Use of HMAC-SHA-1-96 within ESP and AH*, RFC 2404, November 1998.*

12) Department of Defense Instruction, Information Assurance Implementation Draft *No. 8500.bb, September 2001*.

13) The AES Cipher Algorithm and Its Use with IPSec <draft-ietf-ipsec-ciph-aes-cbc.03.txt>, *Internet draft, November 2001.*

14) *Federal Information Processing Standard Publication (FIPS-PUB) 197,* Specification for the Advanced Encryption Standard (AES)*, November 26, 2001*

8.0. TERMINOLOGY

This profile uses a number of terms in specific senses. The following sections provide definitions of the terms that are used. Note that the common criteria term, "users". When used without further clarification, refer to any class of user. Immediately below is are terms used across all PPs to define very basic concepts and ensure that terms are used consistently. This set of common terms is followed by terms used specifically by this PP.

8.1. Common Terminology

Access — Interaction between an entity and an object that results in the flow or modification of data.

Access Control — Security service that controls the use of resources[3] and the disclosure and modification of data.[4]

Accountability — Property that allows activities in an IT system to be traced to the entity responsible for the activity.

Administrator — A user who has been specifically granted the authority to manage some portion or all of the TOE and whose actions may affect the TSP. Administrators may possess special privileges that provide capabilities to override portions of the TSP.

Assurance — A measure of confidence that the security features of an IT system are sufficient to enforce its' security policy.

Asymmetric Cryptographic System — A system involving two related transformations; one determined by a public key (the public transformation), and another determined by a private key (the private transformation) with the property that it is computationally infeasible to determine the private transformation (or the private key) from knowledge of the public transformation (and the public key).

Asymmetric Key — The corresponding public/private key pair needed to determine the behavior of the public/private transformations that comprise an asymmetric cryptographic system.

Attack — An intentional act attempting to violate the security policy of an IT system.

Authentication — Security measure that verifies a claimed identity.

---

[3] Hardware and software.

[4] Stored or communicated.

Authentication data — Information used to verify a claimed identity.

Authorization — Permission, granted by an entity authorized to do so, to perform functions and access data.

Authorized user — An authenticated user who may, in accordance with the TSP, perform an operation.

Availability — Timely[5], reliable access to IT resources.

Compromise — Violation of a security policy.

Confidentiality — A security policy pertaining to disclosure of data.

Critical Security Parameters (CSP) — Security-related information (e.g., cryptographic keys, authentication data such as passwords and pins, and cryptographic seeds) appearing in plaintext or otherwise unprotected form and whose disclosure or modification can compromise the security of a cryptographic module or the security of the information protected by the module.

Cryptographic Administrator — An authorized user who has been granted the authority to perform cryptographic initialization and management functions. These users are expected to use this authority only in the manner prescribed by the guidance given to them.

Cryptographic boundary — An explicitly defined contiguous perimeter that establishes the physical bounds (for hardware) or logical bounds (for software) of a cryptographic module.

Cryptographic key (key) — A parameter used in conjunction with a cryptographic algorithm that determines [7]:

- the transformation of plaintext data into cipher text data,
- the transformation of cipher text data into plaintext data,
- a digital signature computed from data,
- the verification of a digital signature computed from data, or
- a digital authentication code computed from data.

Cryptographic Module — The set of hardware, software, firmware, or some combination thereof that implements cryptographic logic or processes, including cryptographic algorithms, and is contained within the cryptographic boundary of the

---

[5] According to a defined metric.

module.

Cryptographic Module Security Policy **—** A precise specification of the security rules under which a cryptographic module must operate, including the rules derived from the requirements of this PP and additional rules imposed by the vendor.

Defense-in-Depth (DID) **—** A security design strategy whereby layers of protection are utilized to establish an adequate security posture for an IT system.

Discretionary Access Control (DAC) **—** A means of restricting access to objects based on the identity of subjects and/or groups to which they belong. These controls are discretionary in the sense that a subject with a certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject.

Embedded Cryptographic Module **—** One that is built as an integral part of a larger and more general surrounding system (i.e., one that is not easily removable from the surrounding system).

Enclave **—** A collection of entities under the control of a single authority and having a homogeneous security policy. They may be logical, or may be based on physical location and proximity.

Entity **—** A subject, object, user, or another IT device, which interacts with TOE objects, data, or resources.

External IT entity **—** Any trusted Information Technology (IT) product or system, outside of the TOE, which may, in accordance with the TSP, perform an operation.

Identity **—** A representation (e.g., a string) uniquely identifying an authorized user, which can either be the full or abbreviated name of that user or a pseudonym.

Integrity **—** A security policy pertaining to the corruption of data and TSF mechanisms.

Integrity label **—** A security attribute that represents the integrity level of a subject or an object. Integrity labels are used by the TOE as the basis for mandatory integrity control decisions.

Integrity level **—** The combination of a hierarchical level and an optional set of non-hierarchical categories that represent the integrity of data.

Mandatory Access Control (MAC) **—** A means of restricting access to objects based on subject and object sensitivity labels.[6]

Mandatory Integrity Control (MIC) **—** A means of restricting access to objects based

[6] The Bell LaPadula model is an example of Mandatory Access Control

on subject and object integrity labels.

Message Authentication Code (MAC) — A Message Authentication Code is a one-way hash computed from a message and some data. Its purpose is to detect if the message has been altered.

Multilevel — The ability to simultaneously handle (e.g., share, process) multiple levels of data, while allowing users at different sensitivity levels to access the system concurrently. The system permits each user to access only the data to which they are authorized access.

Named Object — An object that exhibits all of the following characteristics:

- The object may be used to transfer information between subjects of differing user identities within the TSF.
- Subjects in the TOE must be able to request a specific instance of the object.
- The name used to refer to a specific instance of the object must exist in a context that potentially allows subjects with different user identities to request the same instance of the object.

Non-Repudiation — A security policy pertaining to providing one or more of the following:

- To the sender of data, proof of delivery to the intended recipient,
- To the recipient of data, proof of the identity of the user who sent the data.

Object — An entity within the TSC that contains or receives information and upon which subjects perform operations.

Operating Environment — The total environment in which a TOE operates. It includes the physical facility and any physical, procedural, administrative and personnel controls.

Operating System (OS) — An entity within the TSC that causes operations to be performed. Subjects can come in two forms: trusted and untrusted. Trusted subjects are exempt from part or all of the TOE security policies. Untrusted subjects are bound by all TOE security policies.

Operational key — Key intended for protection of operational information or for the production or secure electrical transmissions of key streams.

Peer TOEs — Mutually authenticated TOEs that interact to enforce a common security policy.

Public Object — An object for which the TSF unconditionally permits all entities "read" access. Only the TSF or authorized administrators may create, delete, or modify the public objects.

Robustness — A characterization of the strength of a security function, mechanism, service or solution, and the assurance (or confidence) that it is implemented and functioning correctly. There are three levels of robustness:

- **Basic:** Security services and mechanisms that equate to good commercial practices. Basic robustness equates to EAL-2 plus; AMA (Maintenance of Assurance); ALC_FLR (Flaw Remediation), and AVA_MSU.1 (Misuse-Examination Guidance) as defined in CCIB-98-028, Part 3, Version 2.0

- **Medium:** Security services and mechanisms that provide for layering of additional safeguards above good commercial practices. Medium robustness equates to EAL-4 plus; AMA (Maintenance of Assurance); ALC_FLR (Flaw Remediation); ADV_IMP.2; ADV_INT.1; ATE_DPT.2; and AVA_VLA.3 (Moderately Resistant Vulnerability Analysis) as defined in CCIB-98-028, Part 3, Version 2.0. If cryptographic functions are included in the TOE, then the PP should be augmented with AVA_CCA_EXP.2 as documented in the Protection Profile Medium Robustness Consistency Guidance.

- **High:** Security services and mechanisms that provide the most stringent protection and rigorous security countermeasures.

Secure State — Condition in which all TOE security policies are enforced.

Security attributes — TSF data associated with subjects, objects, and users that is used for the enforcement of the TSP.

Security level — The combination of a hierarchical classification and a set of non-hierarchical categories that represent the sensitivity on the information [10].

Sensitivity label — A security attribute that represents the security level of an object and that describes the sensitivity (e.g. Classification) of the data in the object. Sensitivity labels are used by the TOE as the basis for mandatory access control decisions [10].

Split key — A variable that consists of two or more components that must be combined to form the operational key variable. The combining process excludes concatenation or interleaving of component variables.

Subject — An entity within the TSC that causes operations to be performed.

Symmetric key — A single, secret key used for both encryption and decryption in symmetric cryptographic algorithms.

Threat — Capabilities, intentions and attack methods of adversaries, or any circumstance or event, with the potential to violate the TOE security policy.

Threat Agent - Any human user or Information Technology (IT) product or system, which may attempt to violate the TSP and perform an unauthorized operation with the TOE.

User — Any entity (human user or external IT entity) outside the TOE that interacts with the TOE.

Vulnerability — A weakness that can be exploited to violate the TOE security policy.

8.2. Types of Information

Content – Information retrievable through a web server, or executed as a result of information requests to a web server. This includes information requested through Universal Remote Locators (URLs), typically but not limited to Hypertext Markup Language (HTML) files, as well as Common Gateway Interface (CGI) scripts and server side includes.

Server Executable Content – Content that is executed on the server (including but not limited to CGI scripts, Server Side, Active Server Pages, Java Servlets).

8.3. Types of Users

Content Provider – A host system user authorized to provide content and to set access control restrictions on that content. The set of content providers is a subset of the host system users.

Web Server (TOE) Administrator – A host system user authorized to administer the web server. The set of web server administrators is a subset of the host system users, and there are users of the host system that are neither content providers nor web server administrators.

Web Users – A user that accesses the web server using the HTTP or HTTPS through a network port.

8.4. Other Profile Specific Terms

Certificate - A data structure that contains the necessary credentials to authenticate digital signatures and extract session keys from message headers, and that can be determined to be accurate through consultation with a trusted certifying agent. Integrity is usually ensured through the use of strong asymmetric encryption mechanisms.

Content Management Policy - The installation guidelines used by the content providers that will determine acceptable and unacceptable content that will be placed on the web server for use.

9.0. ACRONYMS

The following abbreviations from the Common Criteria are used in this Protection Profile:

ACL	Access Control List
CAPP	Controlled Access Protection Profile
CA	Certificate Authority
CC	Common Criteria
CGI	Common Gateway Interface
CKL	Compromised Key List
CMS	Certificate Management System
CRL	Certificate Revocation List
EAL	Evaluation Assurance Level
GIG	Global Information Grid
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP with a Secure Socket Layer (SSL)
IAFT	Information Assurance Technical Framework
IT	Information Technology
N/A	Not Applicable
PP	Protection Profile
PKI	Public Key Infrastructure
SF	Security Function
SFP	Security Function Policy
SSL	Secure Socket Layer
SOF	Strength of Function

TLS	Transport Secure Layer
TBD	To Be Determined
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSP	TOE Security Policy
WWW	World Wide Web

The following abbreviations from the Common Criteria are used in this Protection Profile:

AES	Advanced Encryption Standard
ATM	Asynchronous Transfer Method
CC	Common Criteria for Information Technology Security Evaluation
DES	Data Encryption Standard
DoD	Department of Defense
DMZ	Demilitarized zone
EAL	Evaluation Assurance Level
ESP	Encapsulating Security Payload
FIPS PUB	Federal Information Processing Standard Publication
FTP	File Transfer Protocol
GIG	Global Information Grid
HTTP	Hypertext Transfer Protocol
I&A	Identification and Authentication
IATF	Information Assurance Technical Framework
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IKE	Internet Key Exchange
IPSEC ESP	Internet Protocol Security Encapsulating Security Payload
IP	Internet Protocol
IT	Information Technology
MRE	Medium Robustness Environment
NBIAT&S	Network Boundary Information Assurance Technologies and Solutions Support
NIAP	National Information Assurance Partnership

NIST	National Institute of Standards and Technology
NSA	National Security Agency
NTP	Network Time Protocol
PKI	Public Key Infrastructure
PP	Protection Profile
RNG	Random Number Generator
SFP	Security Function Policy
SMTP	Simple Mail Transfer Protocol
SOF	Strength of Function
ST	Security Target
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
TOE	Target of Evaluation
TSE	TOE Security Environment
TSF	TOE Security Function
TSP	TOE Security Policy
UDP	User Datagram Protocol
URL	Uniform Research Locator
VPN	Virtual Private Network